

AD-AU97 586

MITRE CORP BEDFORD MA

F/O 9/2

MAR 81 D O CARHOUN, S M WALDSTEIN

F19628-80-C-0001

UNCLASSIFIED

MTR-8122

ESD-TR-81-120

KL

| UF |

AD9758:

1

END

DATE
FILMED

58

DTIC

SD-TR-81-120

MTR-8122

CCD MULTI-LEVEL LOGIC
INTERIM TECHNICAL REPORT

BY D.O. CARHOUN AND S.M. WALDSTEIN

MARCH 1981

Prepared for

RADC SOLID STATE SCIENCES DIVISION
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts



MAR 10 1981

A

Approved for public release; distribution unlimited.

Project No. 6440

Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts

Contract No. F19628-80-C-0001

81 4 10 010

AD A 097586

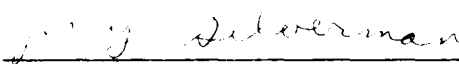
BHC FILE COPY

When U.S. Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

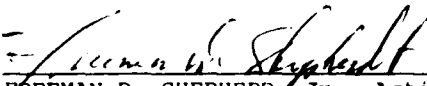
Do not return this copy Retain or destroy.

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.


JERRY SILVERMAN/GS-14
Project Scientist

FOR THE COMMANDER


FREEMAN D. SHEPHERD, Jr., Acting Director
Solid State Sciences Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-81-120	2. GOVT ACCESSION NO. AD-A097586	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) CCD MULTI-LEVEL LOGIC INTERIM TECHNICAL REPORT		5. TYPE OF REPORT & PERIOD COVERED Interim Report Oct 1978 - Dec 1979
7. AUTHOR(s) D. O. Carhoun S. M. Waldstein		6. PERFORMING ORG. REPORT NUMBER MTR-8122
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation P.O. Box 208 Bedford, MA 01730		8. CONTRACT OR GRANT NUMBER(s) F19628-80-C-0001
11. CONTROLLING OFFICE NAME AND ADDRESS RADC Solid State Sciences Division Electronic Systems Division, AFSC Hanscom AFB, MA 01731		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 6440
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 75		12. REPORT DATE MAR 1981
		13. NUMBER OF PAGES 74
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) CHARGE COUPLED DEVICES (CCD) MULTI-LEVEL LOGIC SIGNAL PROCESSING		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) → An interim technical report on Project 6440, CCD multi-level logic documents activities and interim results midway through the project. The report is divided into two principal sections covering (a) analysis and measurements of the practical limitations of CCD multi-level digital operation and (b) ideas and potential techniques for signal-processing applications.		

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

235050

ACKNOWLEDGMENT

This report has been prepared by The MITRE Corporation under Project No. 6440. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
LIST OF ILLUSTRATIONS	5
LIST OF TABLES	7
I. INTRODUCTION	9
1.1 Objectives	9
1.2 Background	9
1.3 Scope	10
II. MULTIPLE-LEVEL CCD OPERATION	11
2.1 Analysis of CCD Multiple-Level Error Rates	12
2.1.1 Solution of the System Equations	17
2.2 Laboratory Measurement of CCD Multiple-Level Error Rates	21
2.2.1 Test Circuitry Description	23
2.2.2 Test Results	25
2.2.3 Continuing Test Plans	28
III. MULTIPLE-LEVEL CCD DIGITAL SIGNAL PROCESSING FUNCTIONS AND OPERATIONAL STRUCTURES	30
3.1 Galois Field Multiplication by Feedback Shift Registers	31
3.2 Galois Field Addition	35
3.2.1 Addition Modulo p	35
3.3 Fast Transform Structures	40
3.3.1 Transform Definition	45
3.4 Cyclic Convolution	60
3.5 Polynomial Division With Transversal Structures	66

TABLE OF CONTENTS (CONCLUDED)

<u>Section</u>	<u>Page</u>
3.6 Galois Field Representation With m-Sequences	69
REFERENCES	74

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	CCD Linear Sequential Circuit Model	16
2	Multi-Level CCD Logic Test Facility	24
3	Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=8$ and $\frac{R_D}{R_S} = 32$	26
4	Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=8$ and $\frac{R_D}{R_S} = 16$	26
5	Input/Output Multi-level Sequence for $R_S=2.5$ MHz, $N=8$ and $\frac{R_D}{R_S} = 8$	27
6	Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $\frac{R_D}{R_S} = 32$	27
7	Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $\frac{R_D}{R_S} = 16$	29
8	Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $\frac{R_D}{R_S} = 8$	29
9	Scaling Multipliers in $GF(5^4)$	32
10	Scaling Multipliers in $GF(3^4)$	34
11	Modulo-p Adder Cell	37
12	4-Input Modulo-p Adder	39
13	8-Input Modulo-p Adder	42
14	Moving Window Discrete Transform	44
15	First Degree Polynomial Divider Structure For An N-Point Discrete Transform:	48

$$x(\ell) = \sum_{k=0}^{N-1} x(k) \alpha^{k\ell}$$

LIST OF ILLUSTRATIONS (CONCLUDED)

<u>Figure</u>		<u>Page</u>
16	Factorization of $x^{80}-1$ over $GF(3^4)$	49
17	80-Point Transform Over $GF(3^4)$ By Spectral Decimation (Schematic)	51
18	Schematic Of An 80-Point Transform Over $GF(3^4)$ By Fast Polynomial Evaluation	55
19	Multiplicative Complexity Of Polynomial Evaluation Discrete Transform Algorithm	58
20	Polynomial Dividers For The Cyclotomic Factors Of $x^{80}-1$	61
21	24-Point Transform Over $GF(5^2)$	62
22	Partitioned Transversal Correlator For A 24-Point M-Sequence Over $GF(5)$ Generated By $x^2 + x + 2$	64
23	Cyclic Autocorrelation Of M-Sequence Over $GF(5)$ Generated By $x^2 + x + 2$	65
24	Polynomial Divider And Its Transposed (Transversal) Form	68
25	Transversal Polynomial Divider For Division By $Q^{(12)}(x) = x^4 - x^2 + 1$	70
26	Finite Field Representation By M-Sequences: Scalar Multiplication And Addition	72

LIST OF TABLES

<u>Table</u>		<u>Page</u>
I.	Intermediate Stored Charge Levels	41
II.	The Elements Of $GF(3^4)$ Generated By $\alpha^4 + \alpha^3 + \alpha^2 - \alpha - 1$	53
III.	Minimal Polynomial Factors Of $x^{80} - 1$ (Splitting Field: $GF(3^4)$)	54
IV.	Multiplicative Complexity Of Discrete Transform Algorithm By Fast Polynomial Evaluation	57
V.	Cyclotomic Factors Of $x^{80} - 1$	59

SECTION I

INTRODUCTION

1.1 Objectives

The principal objective of this project is to evaluate and develop, jointly with RADC/ESE, the concept of using charge coupled devices (CCD's) in the mode of a multiple-level digital processor to perform basic operations of finite-field digital arithmetic as applicable to linear digital signal processing and error correction coding.

1.2 Background

Charge coupled device technology, fundamentally an analog signal processing technology, finds wide and growing application to important signal processing functions such as spectrum analysis, spread spectrum matched filtering and analog storage and integration. It is also an attractive technology for reliable monolithic large scale integration of binary digital logic functions because of its speed of operation, high packing density, low power consumption, and relative simplicity of structures for implementing binary logic.

A charge coupled device, being inherently a sampled analog device, should be capable of operation as a multiple-level digital device if means are provided to detect and refresh the discrete levels being used. Such a capability makes possible the use of CCD's to accomplish the defined operations of addition and multiplication in finite algebraic fields, especially prime number fields. The concept was originally formulated under MITRE IR&D and Technology Base programs in FY'76. That work was reported upon in conceptual application to error correction coding at the 3rd International Conference on the Technology and Application of Charge Coupled Devices [6,7].

Under this project, the previously formulated concepts are being developed further and expanded in application to the general area of finite field digital signal processing. The task efforts emphasize analysis, test and measurement of the multi-level digital signal processing capabilities of state-of-the-art CCD's, leading to device configuration and definition of LSI or VLSI chip architecture to implement defined operations in prime number fields and their extensions. Activities consist of theoretical analysis, laboratory experimentation, test and measurement, demonstration, and documentation. The work will culminate in recommendations for new device development to be undertaken by RADC/ESE.

1.3 Scope

During Fiscal 1979, efforts were applied in the areas of analysis of multiple-level digital error rates, lab measurements of multi-level operation, and the definition and development of processing structures. The accomplishments and status of work in these areas is described in subsequent sections of this report.

This is an interim technical report, documenting activities and results midway through the 30-month effort. The work reported represents an average level of 0.5 MITRE technical staff per month.

The report is divided into two principal sections. Section II describes efforts at analysis and measurements of the practical limitations of multi-level digital CCD operation. Section III describes a number of ideas and potential techniques for signal processing applications of multi-level CCD devices. Since this is an interim technical report and the work is continuing, conclusions and recommendations are reserved for the final technical report, to be published at the conclusion of the 30-month effort.

SECTION II

MULTIPLE-LEVEL CCD OPERATION

A fundamental concern of this project is the ability of typical CCD structures to operate on data consisting of sequences of discrete digital signal charge levels representing signal values that assume one of a finite number of integer levels. Each value may be represented, for example, by an integer multiple of Q_0 elementary charges where Q_0 is the charge difference between readily distinguishable levels in the charge transfer device. Typical operations involved are charge injection, storage, transfer through shift register stages, non-destructive sensing, charge summation, charge splitting, and charge detection. These are the operations to be expected in CCD circuit structures used in multiple-level digital filtering. The ability of the CCD device or circuit to manipulate the charge levels, without modifying them to incorrectly assign the wrong values upon detection, is critically important to the success of the operation. Analog signal processing is more tolerant of noise and distortion introduced by the device. Binary digital signal processing involving Boolean logic operations reduces the problem to one of distinguishing between a pair of levels which can be widely separated to maximize the signal distance relative to the noise. Our work is predicated on the assumption that the noise and distortion introduced by the CCD can be low enough so that the reduced signal distance resulting from use of a larger, yet finite, number of levels can still be adequate to perform useful signal processing functions. It is expected that successful results of such processing methods may be realized as reduced circuit complexity, fewer interconnections, and greater reliability, all obtained with the attendant advantages of simple fabrication, low power consumption, and small feature size. Part of the key to success is to develop innovative uses of the natural CCD operations such as cyclic shifting, transversal filtering, and charge

summation in order to implement the desired processing functions. But first it is necessary to establish the basic ability of the (imperfect) device to perform the essential operations without excessive distortion, and to determine the limits of this mode of operation. We have attempted both analytical and experimental work to answer these questions. The results of our efforts, at this interim stage of the work, are described below.

2.1 Analysis of CCD Multiple Level Error Rates

A theoretical analysis was attempted to determine the average probability of error in estimating the discrete value of a multiple level digital signal observed at the output of a CCD shift register structure. The analytical model included the effects of shift register length (number of stages), charge transfer inefficiency, and intrinsic noise sources. Buried channel device parameters were to be considered. Initially, the analysis was to be based on a Gaussian model of the noise distribution, as implied by use of the central limit theorem. It is realized, however, that for small error probabilities, it is the tail of the distribution that is important and the Gaussian noise model may be inaccurate. Consequently, improved physical and mathematical models of the noise processes are in need of continual examination.

The analysis of error rates should consider not only sampled linear delay lines but also transversal and recursive filter structures. Digital filter structures are commonly described and analyzed by input/output methods. The basic operations performed by a linear filter are often conveniently described in the Z-transform domain in which the filter output is determined as the product of the Z-transform of the input sequence with the system function of the filter, most often given as a rational polynomial in the Z-transform variable. For example, a simple delay line of N stages, each stage suffering a fixed

fractional loss ϵ , has a system function given as

$$H^{(N)}(Z) = \left[\frac{1 - \epsilon}{1 - \epsilon Z^{-1}} \right]^N Z^{-N} \quad (1)$$

which for $\epsilon \ll 1$ can be approximated as

$$H^{(N)}(Z) \approx Z^{-N} \exp (N\epsilon(Z^{-1} - 1)) \quad (2)$$

Such an approach has proven useful in assessing first-order effects of dispersion and frequency response limitations of delay lines (and filters) operating on sampled analog signals. One could next take into account the effect of additive noise (referred to the output or observed as an equivalent output noise) and construct a crude first order model to assess device performance. We have in fact, done this previously in the formulation of a simple Gaussian model, the results suggesting the feasibility of the multi-level logic role for CCD's [1].

In extending the results of such first-order modeling and analysis to transversal and recursive filter structures, the presence of internal feedback loops in the CCD device encumbers the analysis to the extent that the system functions become extraordinarily complicated. Although signal-flow-graph techniques can be readily applied to construct a system function, the computational work involved in reducing it to a practical and usable form does not seem worth the effort, especially in view of a basic theoretical inadequacy of the first-order physical model to accurately describe performance.

A basic problem with the first-order physical model of the sort described is that the random processes that contribute to the charge transfer inefficiency, the recombination charge (dark current) added to each cell, and the uncertainty in sensing the transferred charge at the delay taps are not properly taken into account. In fact in the first order model, the charge transfer loss is not even regarded as a random process variable; only the average value is used as a fixed and

invariant quantity. While the first order model has proven sufficient for describing CCD operation with analog signals, it is doubtful that it can provide an adequate model for accurately predicting digital error rates, especially for the multiple-valued decision with which we are concerned.

Our approach to this problem has been to formulate a dynamical state-variable model of the CCD, viewing it as a linear sequential circuit that can be described by the system equations

$$\underline{x}(k+1) = \underline{A}(k) \underline{x}(k) + \underline{B}(k) \underline{F}(k) \quad (3)$$

$$\underline{y}(k) = \underline{C}'(k) \underline{x}(k). \quad (4)$$

In this formulation the vector $\underline{x}(k)$ represents the state of the model at the k^{th} clock cycle. The matrix $\underline{A}(k)$ describes the unforced operation of the circuit subject only to the initial state. The form of $\underline{A}(k)$ depends on the circuit structure being analyzed (whether a delay line, transversal filter, or recursive filter) and contains the charge transfer loss parameters and also terms involving these parameters in intrinsic feedback loops. In its most general form for our applications $\underline{A}(k)$ is given as

$$\underline{A}(k) = \begin{bmatrix} \epsilon_N(k) & 1-\epsilon_{N-1}(k) & 0 & \dots & 0 & \dots & 0 \\ 0 & \epsilon_{N-1}(k) & 1-\epsilon_{N-2}(k) & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \epsilon_2(k) & 1-\epsilon_1(k) \\ \alpha_1[1-\epsilon_N(k)] & \alpha_2[1-\epsilon_{N-1}(k)] & \dots & \dots & \dots & \epsilon_1(k) + \alpha_2[(1-\epsilon_1(k))] \end{bmatrix} \quad (5)$$

for a structure having N charge transfer cells. $\epsilon_n(k)$ is the fraction of untransferred charge remaining in the n^{th} cell after the k^{th} cycle and $[(1-\epsilon_n(k))]$ is the fraction of charge transferred. $\epsilon_n(k)$ is a random variable resulting primarily from the probabilistic interface-state and bulk charge-trapping phenomena. In the analysis we assume

that this parameter is statistically independent from one cell to another and from one clock cycle to another and that all ensemble-average moments are stationary on k (and from cell to cell). The coefficients $\alpha_1, \alpha_2, \dots, \alpha_N$ are scale factors (or tap weights) weighting the output of each cell in a recursive (LFSR) structure. For a simple delay line, these coefficients would all be set equal to zero. The matrix $\underline{B}(k)$ describes the dependence of the state of the circuit on the external driving sources $\underline{F}(k)$ that in this model include both the input drive and additive noise sources.* The observation matrix $\underline{C}'(k)$ is chosen to express the observed output vector $\underline{y}(k)$ as a function of the circuit state. The linear sequential circuit represented by these equations is sketched in Figure 1.* (The model does not include an output noise source, which can be included in the usual manner).

Our use of a state-space model rather than an input-output description is a departure from what is usually encountered in digital signal processing applications, but it seems necessary because of the intrinsic feedback mechanisms and the further complication of the random processes, which combine to make the model a non-stationary one. As a consequence, the use of the model can be quite complicated. In order to determine the response of the dynamical system described by equations (3) and (4) we must provide an input signal that is typical and then determine the corresponding output by solution of the equations. Since the output sequence (as a function of the index k) will be represented by a sequence of random variables, it is appropriate to calculate at least their first and second order moments in order to calculate error rates. The sufficiency of these moments is based on the assumption that we can treat the output values as Gaussian random variables, for which the mean, variance, and

$$^* \underline{F}(k) = \underline{f}(k) + \underline{v}(k) + \underline{x}(k)$$

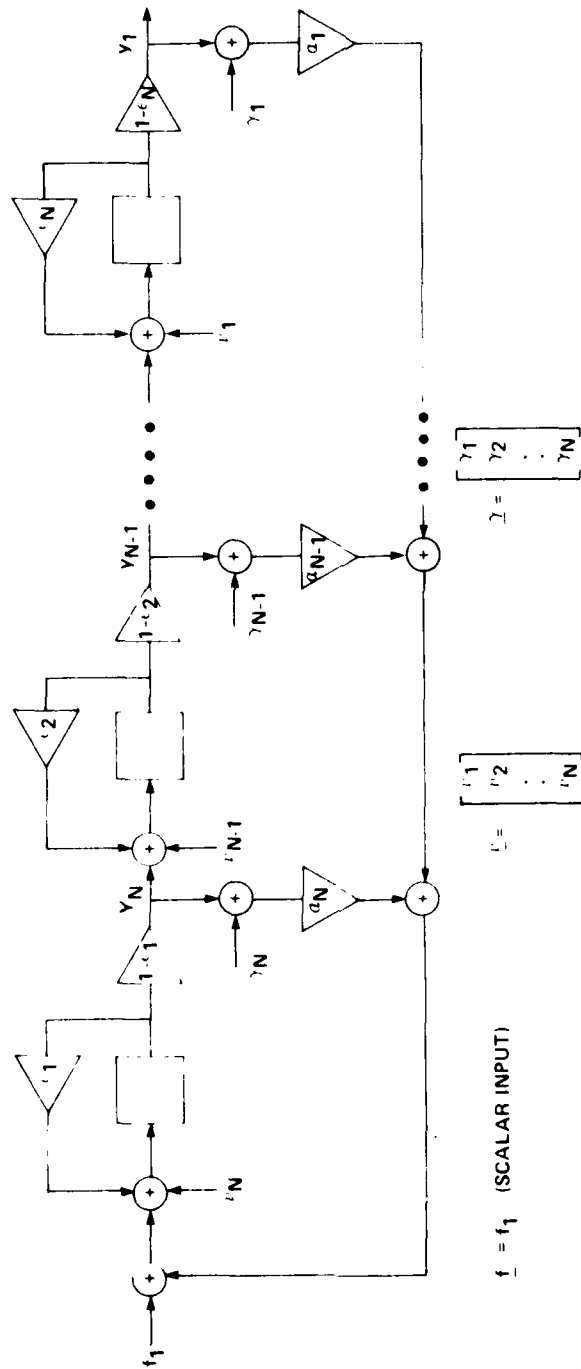


Figure 1. CCD LINEAR SEQUENTIAL CIRCUIT MODEL

covariances completely describe the process.

2.1.1 Solution of the System Equations

A solution to the linear difference equation (3) is given by the variation of constants formula,

$$\underline{x}(k) = \underline{\phi}(k, k_0) \underline{x}(k_0) + \sum_{i=k_0}^{k-1} \underline{\phi}(k, i+1) \underline{B}(i) \underline{F}(i) \quad (6)$$

where $\underline{x}(k_0)$ is the initial state, and the transition matrix $\underline{\phi}(k, k_0)$ satisfies the homogeneous matrix equation

$$\underline{\phi}(k+1, k_0) = \underline{A}(k) \underline{\phi}(k, k_0); \quad \underline{\phi}(k_0, k_0) = \underline{I} \quad (7)$$

and the composition law

$$\underline{\phi}(k, k') \underline{\phi}(k', k_0) = \underline{\phi}(k, k_0); \quad k_0 \leq k' \leq k \quad (8)$$

Direct substitution will verify that equation (7) is solved by the transition matrix

$$\underline{\phi}(k, k_0) = \underline{A}(k-1) \underline{A}(k-2) \dots \underline{A}(k_0) \quad (9)$$

From equation (4), we can express the output as

$$\underline{y}(k) = \underline{C}'(k) \underline{\phi}(k, k_0) \underline{x}(k_0) + \underline{C}'(k) \sum_{i=k_0}^{k-1} \underline{\phi}(k, i+1) \underline{B}(i) \underline{F}(i) \quad (10)$$

with the transition matrix given by equation (9).

Once the output function has been calculated, probabilistic methods can be applied to determine the average probability of incorrectly estimating the output digital level. For example, if we take Q_0 (coulombs) as the maximum (full-well) charge and subdivide it into q equal portions to represent a q -level signal, we can calculate the probability of correct detection (PCD) by integrating the distribution of the output from $Rq - q/2$ to $Rq + q/2$ for a value that is supposed to represent the R^{th} level. The probability of incorrect detection is one minus PCD. If we assume a Gaussian distribution at the output, it is

sufficient to calculate the mean value and variance of $y(k)$ in order to complete the integration (taking account also of additive Gaussian noise at the output). We expect that these parameters of the output distribution will change with the clock cycle k as a result of dispersion and recombination noise increasing with time.

For a solution it is necessary to calculate the ensemble average mean value and variance of equation (10). The work is made tractable by the assumption of statistical independence of the random variables from cell to cell and from cycle to cycle. In particular, we have assumed that

$$\overline{\epsilon_m(k) \epsilon_n(k)} = \overline{\epsilon_m(k)} \overline{\epsilon_n(k)} ; m \neq n \quad (11)$$

and

$$\overline{\epsilon_m(k) \epsilon_m(j)} = \overline{\epsilon_m(k)} \overline{\epsilon_m(j)} ; k \neq j \quad (12)$$

where the overbar indicates an ensemble average.

We assume wide-sense stationarity from cell to cell, so that

$$\overline{\epsilon_m(k)} = \bar{\epsilon} \quad \text{and} \quad \overline{\epsilon_m^2(k)} = \bar{\epsilon^2}, \quad (13)$$

taking $\epsilon_m(k)$ as a random variable of a wide-sense stationary random process. These assumptions allow us to express the ensemble average outputs as

$$\overline{y(k)} = \overline{C'(k)} \overline{x(k)} \quad (14)$$

where

$$\begin{aligned} \overline{x(k)} &= \overline{A(k-1)} \overline{A(k-2)} \dots \overline{A(k_0)} \overline{x(k_0)} \\ &+ \sum_{i=k_0}^{k-1} \left\{ \prod_{j=i+1}^{k-1} \overline{A(j)} \right\} \overline{B} \overline{F(k)} \end{aligned} \quad (15)$$

and where we have assumed statistical independence between the random processes describing the charge transfer inefficiency and the additive noises contributed by the recombination charge and the tap-weight sensing uncertainty. We assume that the latter processes are also wide-sense stationary so that we can express the mean value of the output as

$$\underline{y}(k) = \underline{C}^T \underline{A}^{k-k_0} \underline{x}(k_0) + \underline{C}^T \sum_{i=k_0}^{k-1} \underline{A}^{k-(i+1)} \underline{B} \underline{f}(k) \quad (16)$$

where we have made use of the fact that $\underline{B}(k)$ is a deterministic and constant matrix in our model, and that the average values of $\underline{C}'(k)$ and $\underline{A}(k)$ are constant matrices. Observe that equation (16) could also have been determined by first taking the ensemble averages of equations (3) and (4), making the (ensemble average) state matrix $\underline{A}(k)$ a constant matrix, and then applying the variation of constants formula. Alternatively the solution could be built up by iteration on k . Regardless of the method used, the computation is formidable for practical values of N , suggesting a numerical calculation in place of an analytical approach. Even a direct numerical calculation will be extraordinarily complicated for values of N that are not trivially small, as each iteration requires multiplication by $N \times N$ matrices.

Calculation of the output variance may be performed either by the use of the variation of constants solution of equation (10) in the appropriate statistical formulas in an attempt to develop a closed-form solution, or by developing a difference equation formulation aimed at an iterative numerical calculation. The first approach leads to a complicated closed-form expression that is not

particularly useful. The result of the second approach is a matrix difference equation for the variance,

$$\sigma_x^2(k+1) = \overline{A(k)\sigma_x^2(k)A'(k)} + \overline{[A(k) - \bar{A}(k)] \underline{x}(k) \underline{x}'(k) [A'(k) - \bar{A}'(k)]} + \underline{B} \sigma_f^2(k) \underline{B}' \quad (17)$$

$$\sigma_y^2(k) = \underline{C}'(k) \sigma_x^2(k) \underline{C}(k) \quad (18)$$

where we have expressed

$$\underline{x}(k) \underline{x}'(k) - \underline{\bar{x}}(k) \underline{\bar{x}}'(k) = \sigma_x^2(k) \quad (19)$$

A numerical solution for the variance can be developed from equations (16), (17), and (18) by iteration on k . Numerical calculation of the variance, for practical values of N , will be even more complicated than calculation of the mean value. Terms of the kind $\underline{A}(k) \sigma_x^2 \underline{A}'(k)$ require a pair of $N \times N$ matrix multiplications followed by statistical ensemble averaging of the scalar elements of the product matrix. Clearly, machine-aided computation is necessary. Two approaches suggest themselves:

- 1) Calculation, for small length N and sample value k , on a main frame computer using a suitable programming language (like APL) to perform the matrix operations,

- 2) Use of the state equation directly to simulate the device as a dynamical system on a digital computer, using Monte Carlo techniques for random parameter selection and statistical averaging.

Both approaches for numerical analysis will be practically limited by the available facilities and the computational effort required. They have little advantage over direct measurement of an actual device other than the ability to vary the parameters of the model.

To summarize the status of our analysis, we have concluded that the non-stationarity of the physical model makes inapplicable simple computational techniques, like the Z-transform method, for determination of multiple-level error rates. The techniques of numerical analysis based on a dynamical state-variable model described above seem limited to relatively short structures that can be handled by brute-force machine computation. Since the outcome and ultimate usefulness of such an analysis remain in doubt, we have de-emphasized analysis in favor of experimental measurements. Furthermore, first order models, calculations and previous limited experiments suggest low error rates for a small number, say 3 or 5, of digital levels. Moreover, in considering structures for finite-field operations we conjecture that some very useful operations can be performed in circuits that utilize only a few levels. Some examples will be discussed in Section III.

2.2 Laboratory Measurement of CCD Multiple Level Error Rates

The objective of the laboratory tests and measurements to be described is to assess the ability of charge-coupled devices (CCD's)

to accommodate multiple digital charge levels at low error rates in detecting the valid level. The basic parameter of CCD performance that must be determined is the number of discrete amplitude levels that can be processed and correctly detected for a given device. The percentage of correct detections is the criteria that will be used to compare the performance of different devices. It is also easily related to the error rate which is $1 - \text{PCD}$.

The PCD can be expressed as a function of three basic operating parameters. These three parameters are: the number of discrete levels that exist within the useable dynamic range of the device, M ; the clock frequency of the CCD, f_c ; and the ratio of the input data rate, R_D to the sampling rate, R_S , of the CCD. These operating parameters indirectly affect the PCD which is determined ultimately by the charge transfer inefficiency, dynamic range and intrinsic noise of a given device. Our experimental test facility was designed to enable the PCD to be determined as a function of N , f_c and R_D/R_S .

Different methods of performing laboratory tests and measurements to assess the ability of CCD's to accommodate multiple digital charge levels at low error rate were examined. As a first step in the laboratory program, some effort was expended on careful set-up and operation of a Fairchild CCD-321 dual 455-stage CCD shift register. This is a buried channel CCD that at the time of work represented the best commercially available device of this type. After the operation of the device became thoroughly understood and spurious noise effects were suppressed, an experiment was organized to specifically measure the PCD, or equivalently the error rate, as described below.

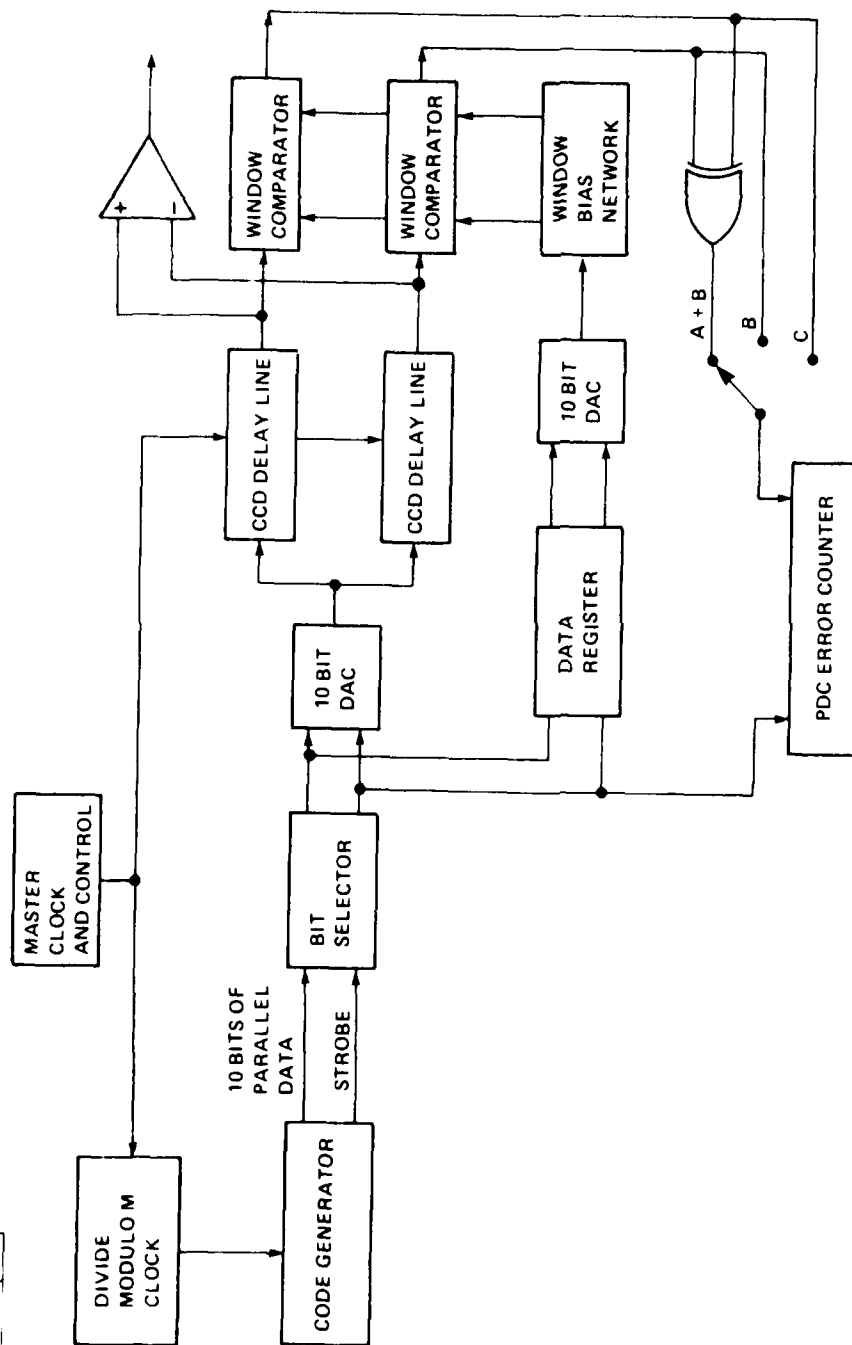
2.2.1 Test Circuitry Description

The test circuitry created to determine the PCD for a given device is modular in form and is capable of adaptation to measurement of most any CCD delay line. A schematic representation of this facility is shown in Figure 2.

The test circuitry consists of a multi-level digital code generator, error detection circuitry, and various control clocks. The multi-level digital signal is derived by digital to analog conversion of the output of a pseudo-random noise (PN) sequence generator. The output word length of this code generator can be varied to change the number of discrete levels present. The PN generator is programmable and controlled by switch selection. The various control clocks allow simultaneous changes in the CCD sampling rate and the data rate. The error detection circuitry is capable of producing both PCD statistics and differential error signals.

The PCD is obtained by comparing the multi-level sequence generated with the sequence present at the output of the CCD delay line. The comparison is made by a window comparator circuit whose interval is determined by the resolution desired. When the output signal is detected correctly, a pulse is generated by the comparator which enables the error counter to accumulate events to determine the PCD.

The error detection circuitry is also capable of comparing two delay lines simultaneously. An Exclusive-OR gate combining the outputs of the two window comparators produces an error signal which determines the number of times the two devices disagree and whether a correct detection was made. A differential amplifier is also used to produce an analog error signal.



2.2.2 Test Results

At the time of writing, the test facility is in the final stages of construction and shakedown. The error detection circuitry, which includes the window comparator, digital delay line, and PCD counter, is 90% complete. The multi-level PN code generator is completely functional and is being used along with a storage oscilloscope to obtain some preliminary data while the test facility is being completed.

Two CCD's commercially available (from Fairchild and Reticon) have been obtained for testing purposes. These analog delay lines appear to represent the best commercially available devices of this type but are not as suitable as other devices under development (RCA's RSAM for example). The CCD presently under test is the CCD-321A video delay line produced by Fairchild. This device contains two buried channel 455-stage analog shift registers. However, it requires the inconvenience of four-phase clocking.

Several multi-level digital sequences with $M = 8$ discrete levels were sampled by the delay line at a sample rate $R_s = 2.5$ MHz. The data rate R_D of the generated sequence was then varied to produce different ratios of oversampling. Photographs for R_s/R_D ratios of 32, 16, and 8 can be found in Figures 3, 4, and 5 respectively. The change in the data rate (versus a constant sampling rate) is shown by the change in the horizontal time axis of each photograph.

It can be seen from these photographs that the changes in data rates within the range examined seem to have little effect on the transmission of data through the CCD delay line. However, one can see that the presence of clock feedthrough on the output signal will probably cause errors to occur in the detection process.

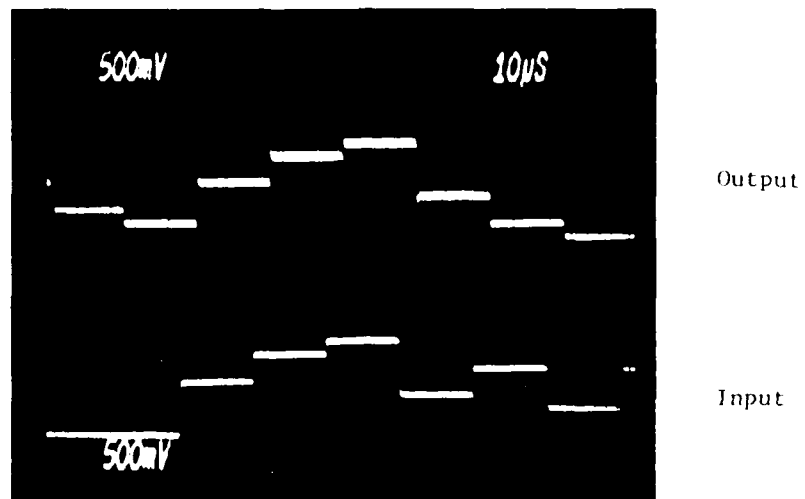


Figure 3. Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=8$ and $R_S/R_D = 32$

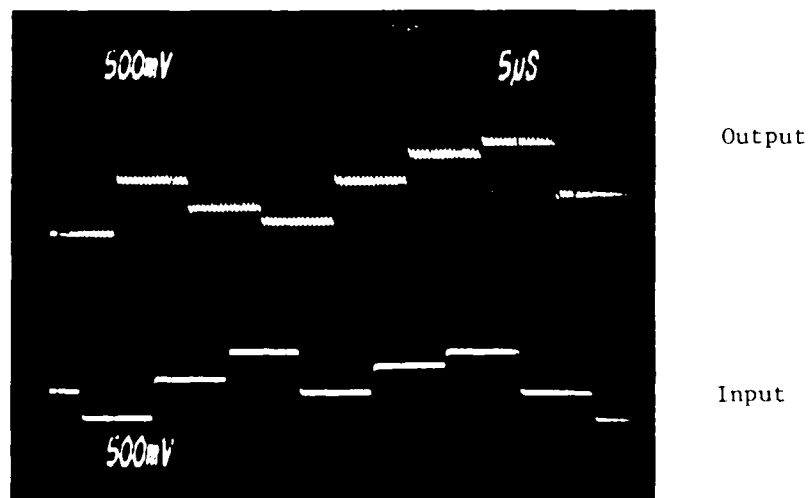


Figure 4. Input/Output Multi-level Sequence for $R_S=2.5$ MHz, $N=8$ and $R_S/R_D = 16$

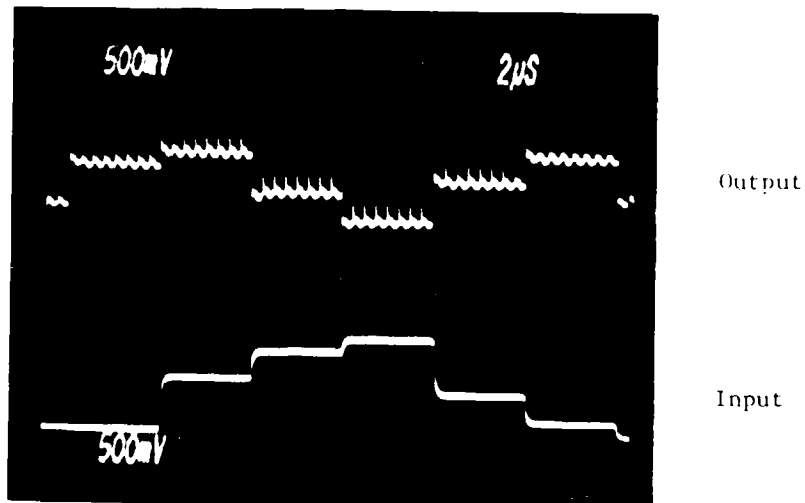


Figure 5. Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=8$ and $R_S/R_D = 8$

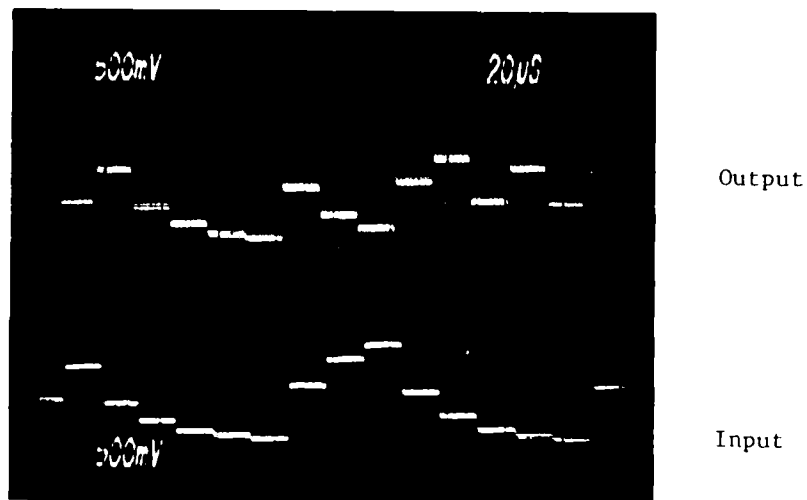


Figure 6. Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $R_S/R_D = 32$

The effects of the ambiguities created by the clock feedthrough can be clearly seen in Figures 6, 7, and 8. This noise source causes two sequential levels to overlap as can be seen in Figure 6. This ambiguity is present for $M = 32$ and R_S/R_D values of 32, 16, and 8. Several techniques are being explored to eliminate this source of noise. The two most promising methods are additive cancelling of the clock and lowpass filtering. Both techniques will be used.

2.2.3 Continuing Test Plans

The testing of the Fairchild, Reticon, and other available CCD's is ongoing. At this time, we are concentrating our resources toward the completion of the test facility. While the test facility is being completed, we are probing the optimum operation of the devices being tested. The parameters of charge transfer efficiency, dynamic range, and frequency response are being determined for each device. These preliminary device evaluations should help us to better understand the operation of each as a multi-level digital delay line. The procedures developed will help determine the optimum range of the input and clock biasing for proper multi-level operation.

After completion of the test circuitry, statistical data will be gathered on the performance of each device when used to process multi-level digital signals. The probability of correct detection statistics will be examined and plotted as a function of M , f_c , and R_S/R_D . We will determine what effects charge transfer efficiency, dynamic range, and the number of device stages have on the PCD for a given device with the results analyzed and applied to more complicated processing structures.

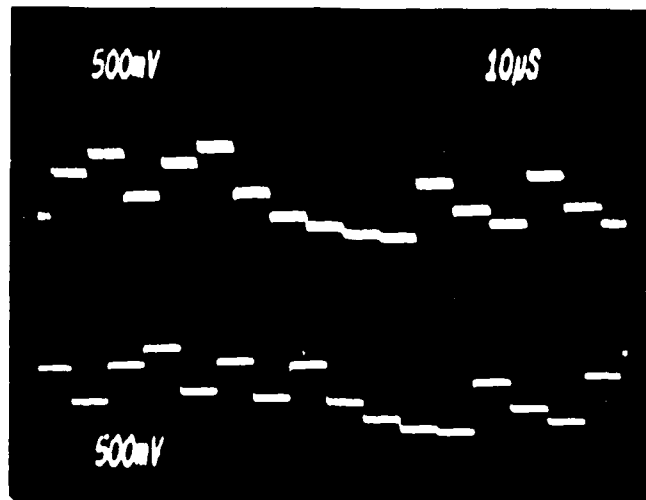


Figure 7. Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $R_S/R_D = 16$

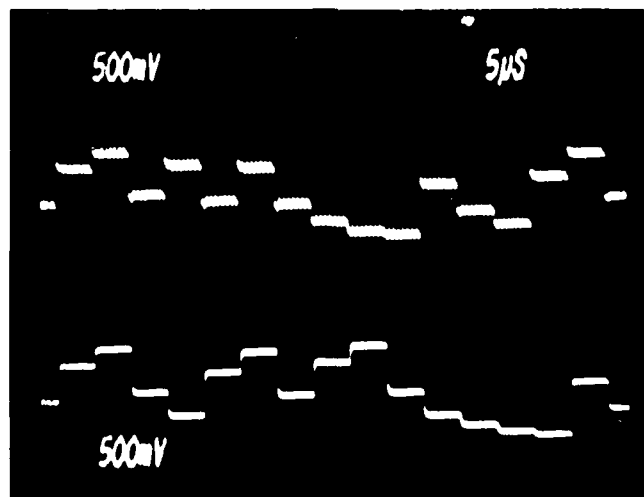


Figure 8. Input/Output Multi-Level Sequence for $R_S=2.5$ MHz, $N=32$ and $R_S/R_D = 8$

SECTION III

MULTIPLE-LEVEL CCD DIGITAL SIGNAL PROCESSING FUNCTIONS AND OPERATIONAL STRUCTURES

Given that multi-level error rates for state-of-the-art CCD's are sufficiently low, we must still devise efficient monolithic structures to perform the needed operations. Work elsewhere is concerned with the use of CCD's for multiple-valued logic operations based on extended Boolean logic. Our work is based on operations in finite algebraic fields or rings for which circuitry needs to be developed to carry out the basic algebraic operations of addition, multiplication, and inversion. We previously observed that prime-field multiplication can be performed by cyclic permutation of the multiplicative group of the field and that, similarly, addition can be carried out by cyclic permutation of the additive group [1]. But most signal processing functions carried out in finite fields will require extension-field operations to be performed, equivalent to operating with polynomials defined over the prime field. Although two-dimensional array multipliers organized in binary trees have previously been advocated for standard elements to carry out the operation, our view of the approach is that it tends to expand the hardware complexity [2]. This approach has the attendant risk of decreased circuit reliability and increased cost, compensated by the ease of field-programmability and the potential for incorporating some degree of fault-tolerance through structural redundancy. But basically the finite-field array multiplier approach seems not well suited to typical CCD operations, although the option should be kept open for further exploration.

Below we discuss some of our ideas for carrying out Galois field operations with reference to the typical signal processing operations of discrete transformation and cyclic convolution. Our object is to

devise structures in which the natural CCD functional operations can be used to advantage; consequently there is some emphasis on shift-register-like structures. Our ideas at this stage are tentative and exploratory, and certainly in need of further development (or selective abandonment). They are also intended to suggest some useful test structures for exploratory device development and fabrication.

3.1 Galois Field Multiplication by Feedback Shift Registers

Earlier work showed that computations in the base field $GF(p)$ could be performed by cyclic permutation of the elements of the additive or multiplicative groups of the field and simple circuitry using the Fairchild CCD-311 was configured to demonstrate the principle for $p = 5$ [1]. It was largely this result that prompted us to investigate further the capacity of a CCD to unambiguously store and manipulate charge samples that represent distinct elements of $GF(p)$. Similar techniques can be used for the extension-field operations.

It is well known that multiplicative operations in $GF(2^m)$, such as scaling by a fixed element α^k of $GF(2^m)$, raising to powers $(\alpha^k)^r = \alpha^{kr}$, and multiplying two variables $\alpha^k \alpha^l$, can be performed by linear sequential circuits in which the arithmetic operations are carried out in the prime field $GF(2)$. Under this project we have examined the generalization to $p \neq 2$ with the result that similar circuits can be devised in $GF(p^m)$ where $p \neq 2$ is a prime number. For example, it is possible to multiply an element α^k of $GF(p^m)$ by a fixed element (α^l) by shifting the data sample α^k (once) in an m -stage linear sequential circuit whose feedback and feedforward connections are determined by the scale factor α^l . The connection matrix for the circuit can be determined easily from the field-generating recursion, which the matrix must also satisfy, with the result that it can be written down by inspection. As an example, we have drawn in Figure 9 a set of shift registers that can be used for multiplying by the elements of $GF(5^4)$; only a few are actually shown

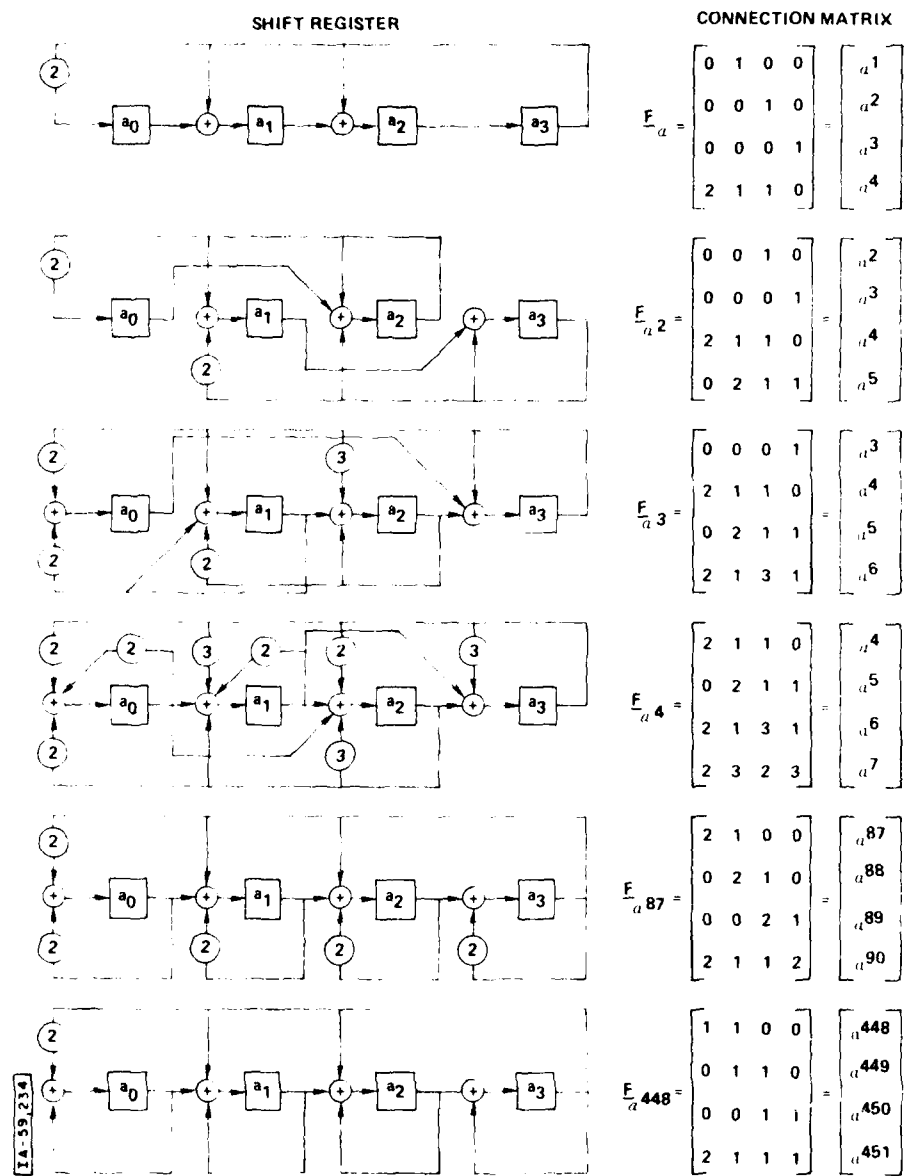


Figure 9. SCALING MULTIPLIERS IN $GF(5^4)$

for purposes of illustration. Each of these circuits forms the required product as the contents of the register in a single data shift. The operations of addition and multiplication are carried out in the prime base field (modulo 5). In addition to defining sums in $GF(5)$, it is also necessary in the structures shown to implement scalar multiplication by the elements of $GF(5)$.

If we work with an extension field of the form $GF(3^m)$, then the operations of addition and multiplication of the base field elements are further simplified since the elements of $GF(3)$ can be represented as 0, 1, -1. Consequently, non-zero multiplication is achieved either by sign inversion or non-inversion of the signal. A set of multipliers that implement scalar multiplication by the elements α^k of $GF(3^4)$ is shown in Figure 10 using this representation for the elements of the base field.

Since there are $p^m - 1$ non-zero elements in $GF(p^m)$ -- 80 elements for $GF(3^4)$ -- one might expect to require the same number of registers to multiply data by all the field elements in parallel in a single clock cycle. Actually only $m-1$ linearly independent registers would be necessary to generate all of the products providing their outputs are appropriately combined. Notice for example that:

$$\underline{F}_{-\alpha}^{67} = \underline{F}_{-\alpha}^1 + \underline{F}_{-\alpha}^2 \quad \text{and} \quad \underline{F}_{-\alpha}^{68} = \underline{F}_{-\alpha}^2 + \underline{F}_{-\alpha}^3 \quad (20)$$

so that the results of multiplying by α^{67} can be obtained by multiplying separately by α^1 and α^2 and adding the products. Also, the existence of unique multiplicative inverses can be used to reduce the number of separate registers; for example $\alpha^{42} = \alpha^{2 \cdot 42}$ or equivalently $\underline{F}_{-\alpha}^2 = \underline{F}_{-\alpha}^{42}$ so that multiplication by α^{42} can be accomplished by first multiplying by α^1 and then complementing the output. In this way each pair of registers can be used to generate at least 5 products on one clock cycle.

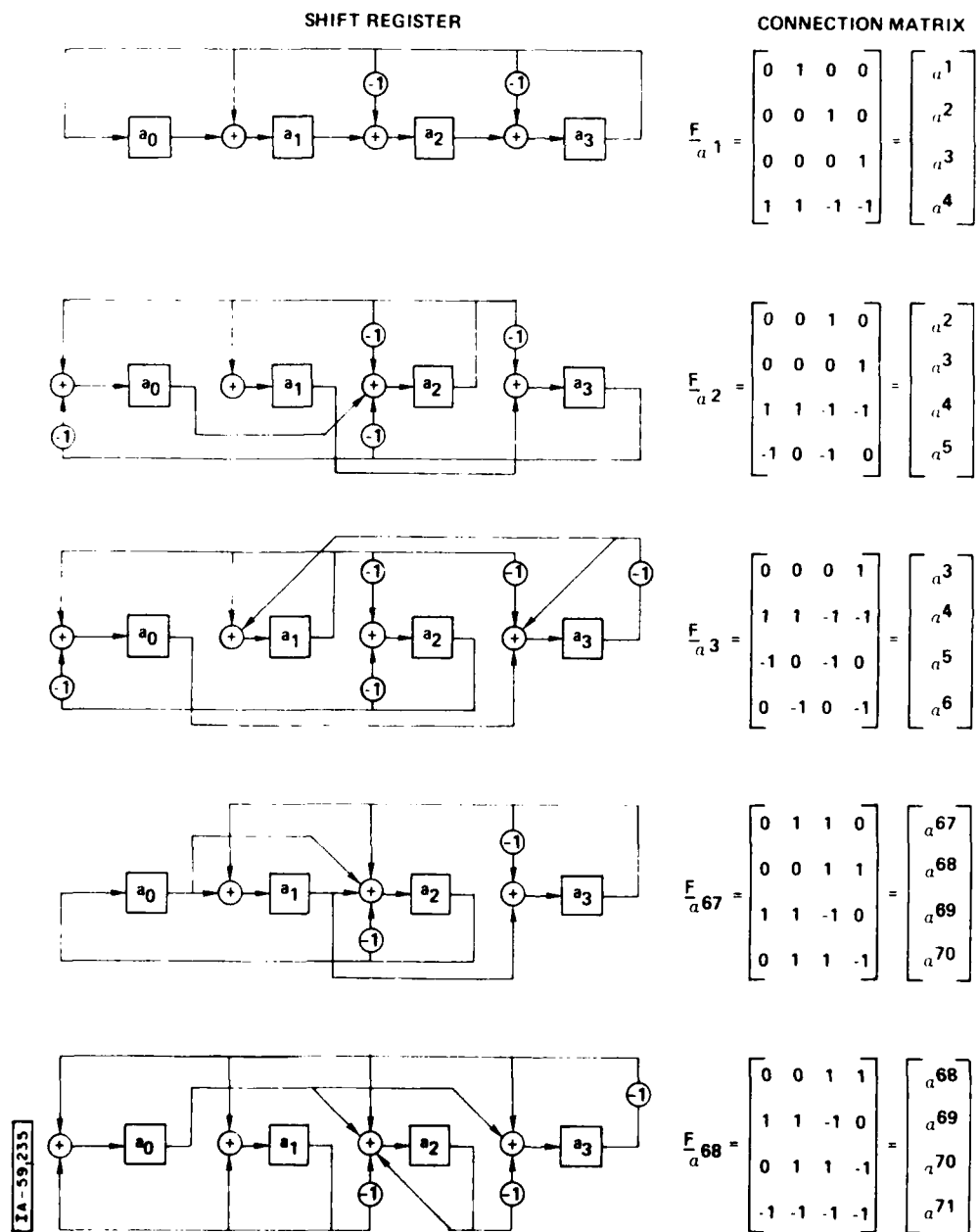


Figure 10. SCALING MULTIPLIERS IN $GF(3^4)$

3.2 Galois Field Addition

Addition in $GF(p^m)$ may be considered as the addition of polynomials of degree $m-1$ having coefficients in the prime field $GF(p)$. The addition is carried out by adding (modulo p) the coefficients of the variables of the same degree. Unlike the addition of binary n -tuples corresponding to radix 2 numerical representation, there is no carry operation. The operation is the same as cartesian addition of m -dimensional vectors.

3.2.1 Addition Modulo p

One of the most important functions that needs to be developed is addition modulo p . As discussed previously, we can treat addition in $GF(p^m)$ as m -vector addition over $GF(p)$ in which the vector components are added modulo p . Multiplication in $GF(p^m)$ can be implemented either by exploiting the cyclic property of the multiplicative group (as shown in 3.1 above) or by performing serial multiplication in which partial products (modulo p) are formed and then added (or accumulated) by vector addition modulo p as in the case of an array multiplier. No matter how we partition the computation over $GF(p^m)$, it is inescapable that $GF(p)$ adders will be required. Such an adder can be developed by extending the notion of CCD digital logic techniques employed by TRW for performing binary logic operations [3]. One such scheme for a $GF(p)$ adder is described below.

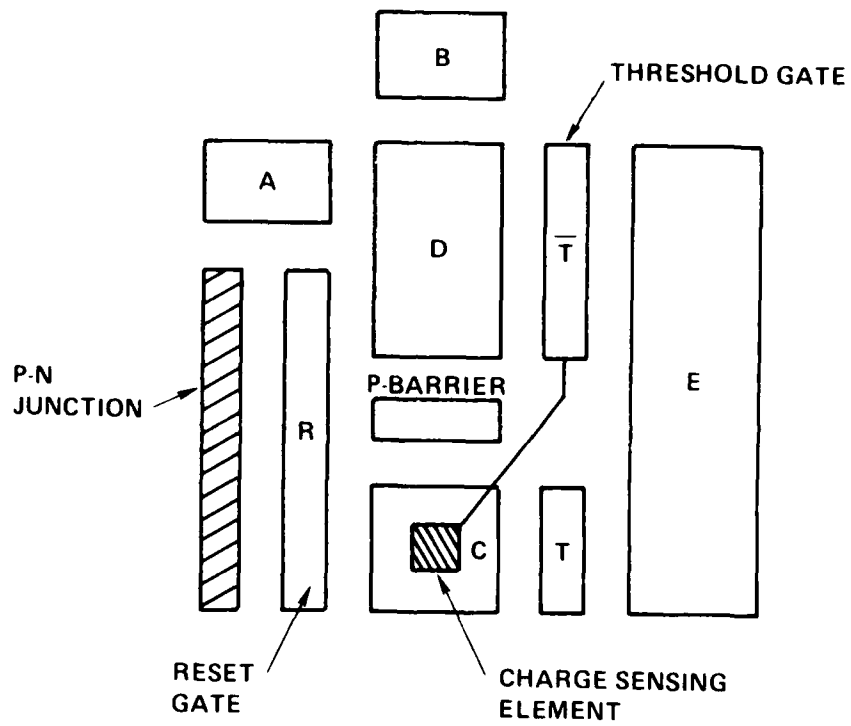
The operation of addition modulo p for a pair of elements a, b , of $GF(p)$ is defined by the simple rule:

$$(a + b)_{\text{mod } p} = \begin{cases} a + b; & a + b < p \\ (a + b) - p; & a + b \geq p \end{cases} \quad (21)$$

where the operations on the right hand side of equation (20) are the operations normally defined in the infinite field of all the integers. A CCD structure that executes this operation is shown in Figure 11; it represents a slight variation of the cellular structure used by TRW for a binary Exclusive-OR gate [3]. The operation of the suggested adder can be described by the following sequence of events, involving charges that exceed the bias (zero) level.

1. On the first clock pulse, charge packets stored under electrodes A and B are transferred and combined under electrode D. The charge exceeding the controlled value p flows over the barrier and accumulates under electrode C.
2. On the next pulse, the charge packets residing either under gate C or gate D are transferred to the region under electrode E, depending on the states of the transfer electrodes T and \bar{T} . The \bar{T} electrode is controlled by the element sensing the charge under electrode C the presence of charge under C inhibiting the transfer from D. The charge under C is transferred to E in either case, being either zero or data.
3. On the subsequent pulse the charge on electrode E is sensed and the electrodes C and D are preset to the zero level by transferring their remaining charge to a diode charge sink. The charge packets representing the next set of values to be added are transferred to electrodes A and B and the cycle is ready to repeat.

If $a + b \geq p$ then $(a + b)_{\text{mod } p}$ is transferred to electrode C during the first third of the cycle and is transferred to electrode E in the second step. The charge remaining under D must equal the modulus value p , but is prevented from further transfer due to the charge present under C. If $a + b < p$ then no charge is transferred to electrode C (by overflowing the barrier) and the value $(a + b)_{\text{mod } p}$ resides under D after the first step, and is transferred from D to E on the second step, the transfer gate \bar{T} now permitting the transfer since no charge is sensed under C. On the third part of the cycle, the residue



IA-59,230

Figure 11. MODULO-P ADDER CELL

$(a + b)_{\text{mod } p}$ is sensed on electrode E and the other gates are re-initialized.

The technique described can be modified to configure a 4-input adder. The operation of such a structure, shown in Figure 12, can be described by the following sequence of operations:

1. Charge packets stored under electrodes A_1 and B_1 are transferred to electrode C_1 ; charge in excess of the modulus value p is allowed to flow over the controlled barrier and accumulate under E. Simultaneously, the charge packets under A_2 and B_2 are transferred to C_2 with the charge in excess of p allowed to flow over the barrier and accumulate under E. Any charge accumulated under E that exceeds the modulus value p is allowed to cross the barrier and accumulate under F.
2. After completion of step (1) any charge packets residing under C_1 are transferred to C_2 by enabling the appropriate transfer gate. Again, charge in excess of the modulus value flows over the barrier to E, and any charge in E that exceeds p flows over the subsequent barrier and settles under electrode F.
3. In the next step, the charge under E is sensed to either permit or inhibit the transfer of charge from C_2 to E, the presence of non-zero signal charge in E inhibiting the transfer.
4. In the final step, the charge under F is sensed to either permit or inhibit the transfer of charge from E to F by control of the transfer gate. Non-zero signal charge in F prohibits the transfer. The charge under F at the completion of this step is sensed to determine the value $(A_1 + B_1 + A_2 + B_2) \text{ modulo } p$.

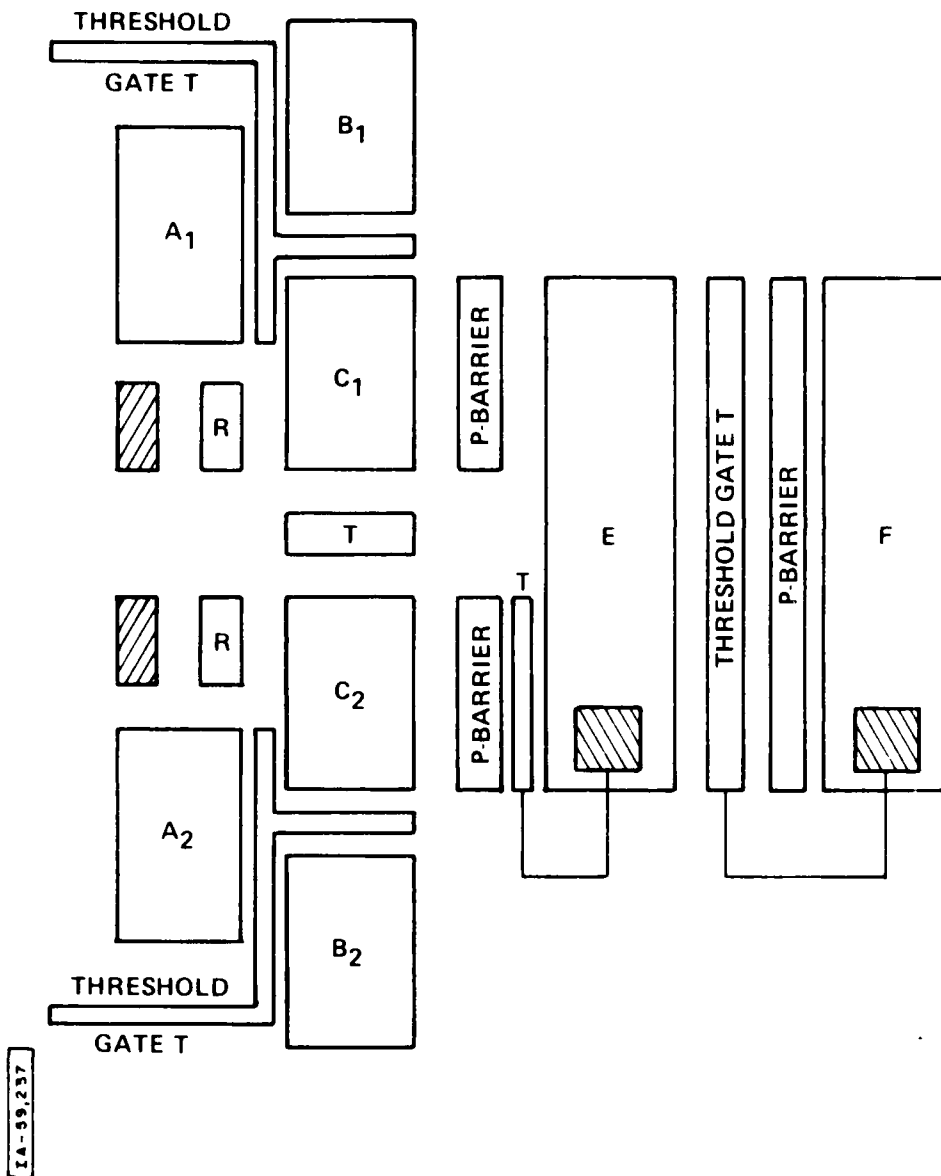


Figure 12. 4-INPUT MODULO-P ADDER

In order to verify the correct operation of the scheme just described, it is convenient to display the various levels of charge stored under the electrodes after the second step. These are dependent on the values of the summed charge packets as shown in Table 1. The last condition motivates step 3 above; afterwards the sum is stored either under electrode E or F and is sensed at the completion of step 4.

The scheme outlined can be developed into an 8-input adder by providing an additional charge accumulating cell and controlled barrier. Such an adder is shown schematically in Figure 13.

The adders described schematically are presented as exploratory ideas. The actual details of clocking, formation of potential barriers, and sensing techniques need to be examined more closely in order to assess the feasibility of the scheme.

3.3 Fast Transform Structures

The work being carried out under this project was motivated originally by the prospect of devising simple structures, based on multi-level CCD operation, for decoding Reed-Solomon error-correcting codes. It was previously established that such codes could actually be designed over $GF(p)$ where p is a prime number greater than 2. The advantage seen was that the arithmetic operations would be performed in the base field $GF(p)$ rather than in some extension field $GF(p^m)$ with the result that the hardware could be simplified if CCD multi-level digital processing could be used. In order to make the codes useful it would be necessary for p to be reasonably large, say $p = 17$ or $p = 31$, thus prompting the examination of CCD operation with such numbers of discrete amplitude (charge) levels.

Lately, we have come to believe that it may be more useful to work in an extension field where both the characteristic p and the degree of extension m are small. For example, we might choose

TABLE I
INTERMEDIATE STORED CHARGE LEVELS

CONDITION*	C	C ₂	E	F
$A_1+B_1 \geq p, A_2+B_2 \geq p$ and $[A_1+B_1]_p + [A_2+B_2]_p \geq p$	ϵ	p	p	$[A_1+B_1+A_2+B_2]_p$
$A_1+B_1 \geq p, A_2+B_2 \geq p$ and $[A_1+B_1]_p + [A_2+B_2]_p < p$	ϵ	p	$[A_1+B_1+A_2+B_2]_p$	ϵ
$A_1+B_1 \geq p, A_2+B_2 < p$ and $[A_1+B_1]_p + [A_2+B_2]_p \geq p$	ϵ	p	p	$[A_1+B_1+A_2+B_2]_p$
$A_1+B_1 \geq p, A_2+B_2 < p$ and $[A_1+B_1]_p + [A_2+B_2]_p < p$	ϵ	p	$[A_1+B_1+A_2+B_2]_p$	ϵ
$A_1+B_1 < p, A_2+B_2 \geq p$ and $[A_1+B_1]_p + [A_2+B_2]_p \geq p$	ϵ	p	$[A_1+B_1+A_2+B_2]_p$	ϵ
$A_1+B_1 < p, A_2+B_2 \geq p$ and $[A_1+B_1]_p + [A_2+B_2]_p < p$	ϵ	p	p	$[A_1+B_1+A_2+B_2]_p$
$A_1+B_1 < p, A_2+B_2 < p$ and $[A_1+B_1]_p + [A_2+B_2]_p \geq p$	ϵ	p	$[A_1+B_1+A_2+B_2]_p$	ϵ
$A_1+B_1 < p, A_2+B_2 < p$ and $[A_1+B_1]_p + [A_2+B_2]_p < p$	ϵ	$[A_1+B_1+A_2+B_2]_p$	ϵ	ϵ

* $A_i + B_i = [A_i + B_i]_p$ modulo p
 ϵ = bias charge only

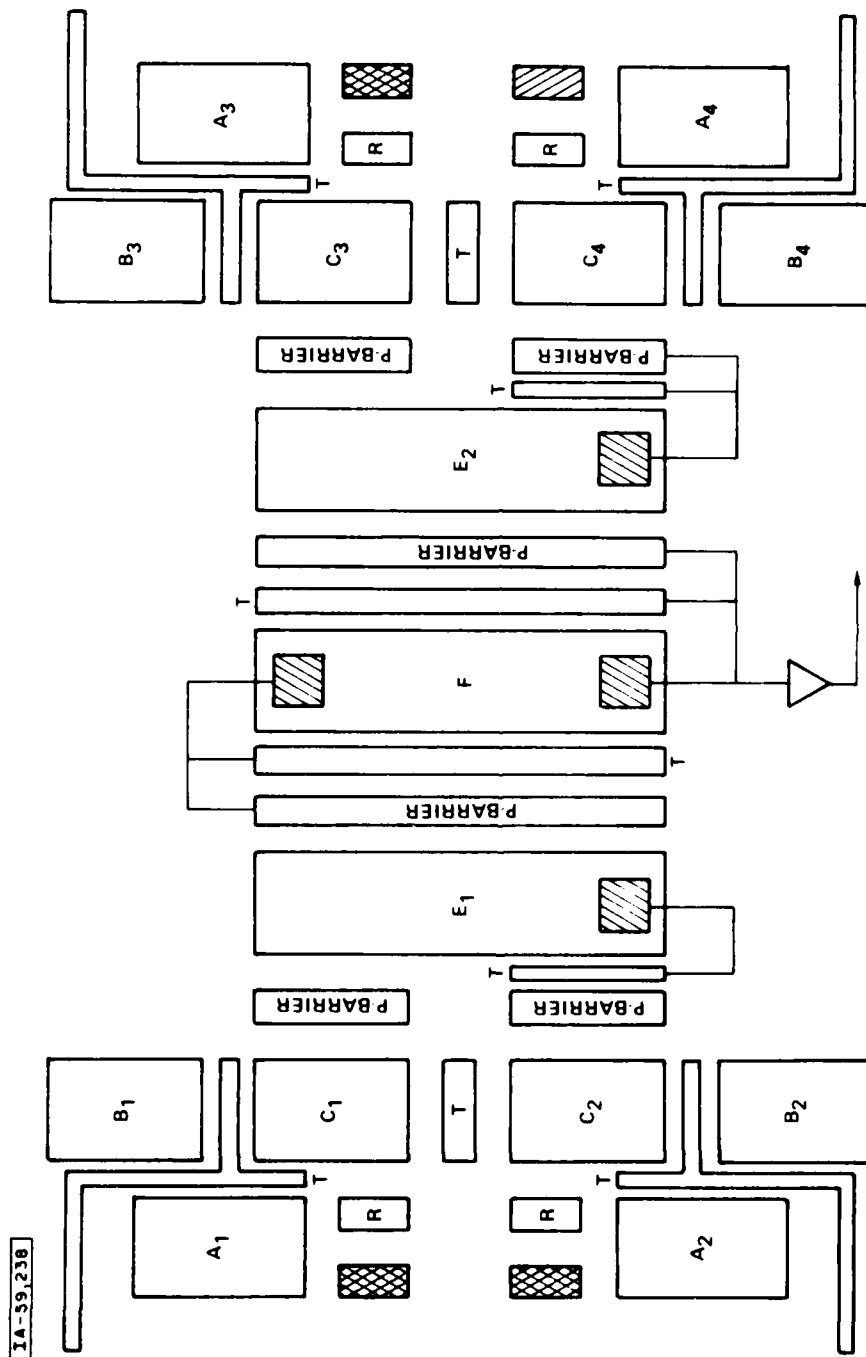


Figure 13. 8-INPUT MODULO-2 ADDER

$p = 3$ and $m = 4$ to design a Reed-Solomon code having a block length of $p^m - 1 = 80$ symbols of 3 levels each. Our change in direction is prompted by several factors. First of all it is becoming apparent that the extension field operations are not overly complicated when the field characteristic is small, as was discussed above for the multiplier structures. Secondly, the requirements for multi-level CCD operation are reduced to levels for which high reliability is evident. Finally, the use of such extension fields admits the use of fast transform techniques that can be effectively employed in a Reed-Solomon decoding algorithm, and probably in other digital signal processing applications as well.

A discrete transform can be defined over $GF(p^m)$ that is analogous to the discrete Fourier transform (DFT) defined over the field of complex numbers [4]. This transform is interesting in its own right for conceptual reasons and also because it exhibits the cyclic convolution property which can be useful to evaluate the convolution of two sequences by transform techniques in which the product of the transforms produces the transform of their convolution. The other well-known Fourier transform properties are also useful computationally.

In addition, the discrete Fourier transform is strongly linked with the realization of digital filters that implement a rational transfer function [5]. More recently, techniques of finite algebra have been applied to the design of digital filters in a manner that overcomes some of the limitations (approximation error, roundoff error, instability) of digital filter design [6]. One rather general approach to finite-field digital filter synthesis realizes the filter (in each field representation) as the weighted sum of the coefficients of the moving window discrete transform of the input, as shown schematically in Figure 14.

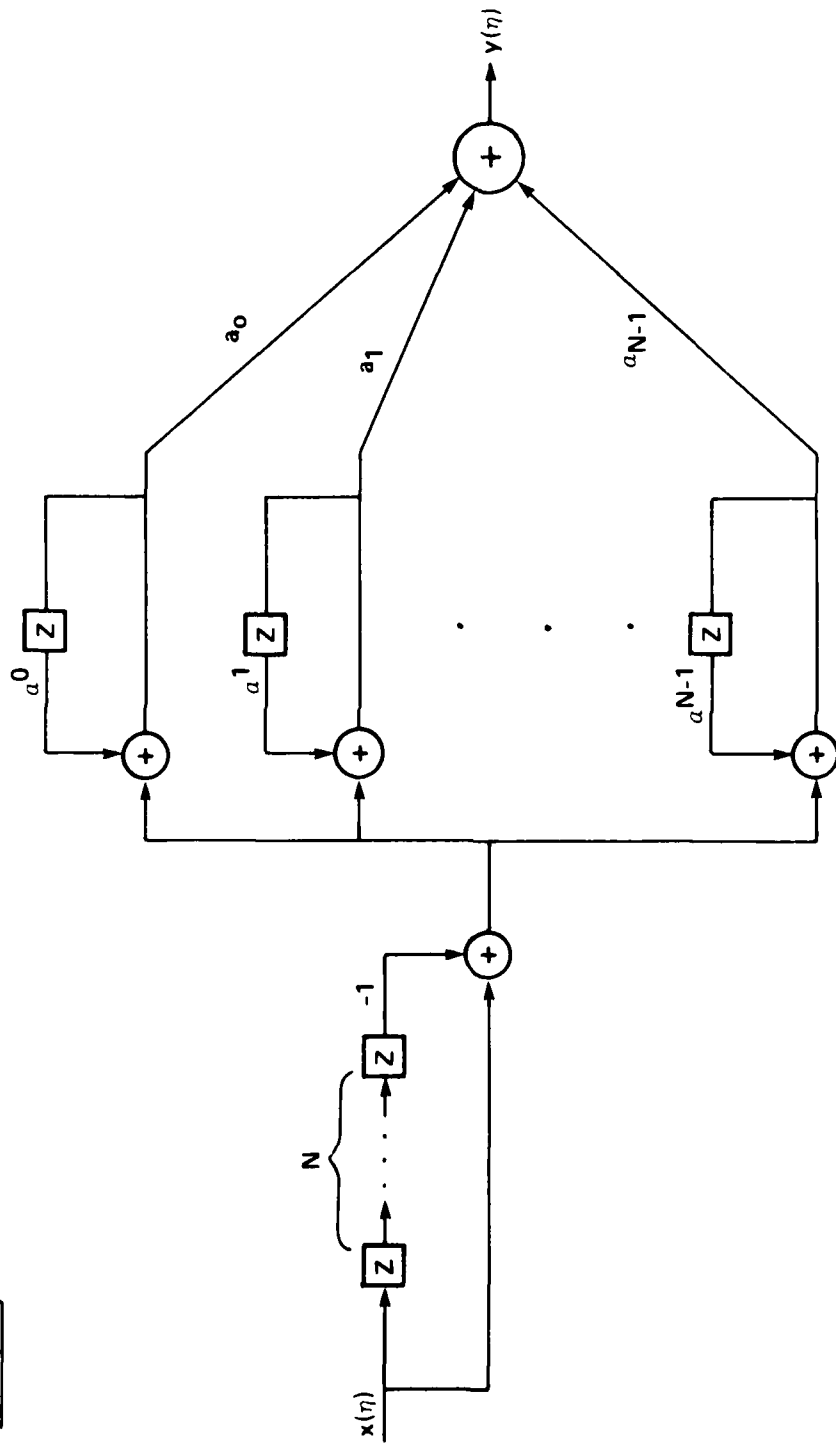


Figure 14. MOVING WINDOW DISCRETE TRANSFORM

Sensing the importance of the discrete transform function, we have expended some effort on this project to examine structures that implement the transform in the class of Galois fields $GF(p^m)$ where $p > 2$ is a prime number. We have found that a systematic fast computational algorithm can be devised that, unlike the Winograd algorithm, applies systematically to all such fields. This led us further to examine the processing structures implied and the implications of the required arithmetic operations with regard to the use of multi-level CCD techniques. Some results of this work are described below, with the inclusion of a specific example for clarity.

3.3.1 Transform Definition

Let a_0, a_1, \dots, a_{n-1} be distinct elements of a finite algebraic field $GF(p^m)$ of order p^m-1 having an element b of order n . The linear transformation

$$A_j = \sum_{i=0}^{n-1} a_i b^{ij} \quad (22)$$

is a mapping of $GF(p^n)$ into itself. It is assumed that n divides p^m-1 , the order of the field, and for our purposes will be equal to it. In that case the field element b is a primitive n^{th} root of unity. It can be shown for any integer r ,

$$\sum_{i=0}^{n-1} b^{ir} = \begin{cases} n, & r \equiv 0 \pmod{n} \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

and the property can be used to verify by direct calculation that the mapping that is inverse to that of equation (22) is the linear transformation

$$a_i = n^{-1} \sum_{j=0}^{n-1} A_j b^{-ji} \quad (24)$$

where $n^{-1}n = p^m - 1$. Equations (22) and (24) define a discrete transform pair over $GF(p^m)$ and the operations of addition and multiplication are defined in the same field. Addition may be performed as modulo- p addition of the m -tuples that are the field elements comprising the sum. Multiplication may be defined by addition of indices of the field elements

$$b^r b^s = b^{r+s}. \quad (25)$$

The transform pair of equations (22) and (24) are analogous to the discrete Fourier transform pair for which b would be a complex n^{th} root of unity and the arithmetic would be defined in the complex number field: in particular, the cyclic convolution property holds. Fast computation algorithms, analogous to the FFT algorithms, can also be applied.

If the sequence to be transformed is expressed as a polynomial over $GF(p^m)$

$$a(x) = a_0 + a_1 x + a_2 x^2 + \dots + a_{n-1} x^{n-1} \quad (26)$$

then the transform of the sequence $a_0, a_1, a_2, \dots, a_{n-1}$ is seen to be identical with polynomial evaluation of $a(x)$ at the n distinct points $b^0, b^1, b^2, \dots, b^{n-1}$ and the inverse transform is identical with interpolation of the polynomial $a(x)$ from its n values.

$$a(b^j) = a_0 + b^j (a_1 + \dots + b^j (a_{n-2} + b^j a_{n-1}) \dots) \quad (27)$$

or equivalently it can be interpreted as the remainder of the polynomial division $a(x)/(x - b^j)$ evaluated at the point b^j . The second interpretation may be represented as the set of polynomial congruences,

$$a(x) \equiv a(b^j) \pmod{x - b^j}; \quad j = 0, \dots, n-1. \quad (28)$$

The congruences of equation (28) can be calculated in principle by dividing the polynomial $a(x)$ separately by the first degree polynomials $(x - b^j)$, keeping only the remainders. That is operationally equivalent to evaluating $a(x)$ at the n non-zero field points b^j . In either case n^2 multiplications in $GF(p^m)$ are implied.

A class of fast computational algorithms--fast because they reduce the number of multiplications in $GF(p^m)$ --can be devised by consideration of the different ways of factoring the polynomial $x^n - 1$ over $GF(p^m)$. One way is to factor as

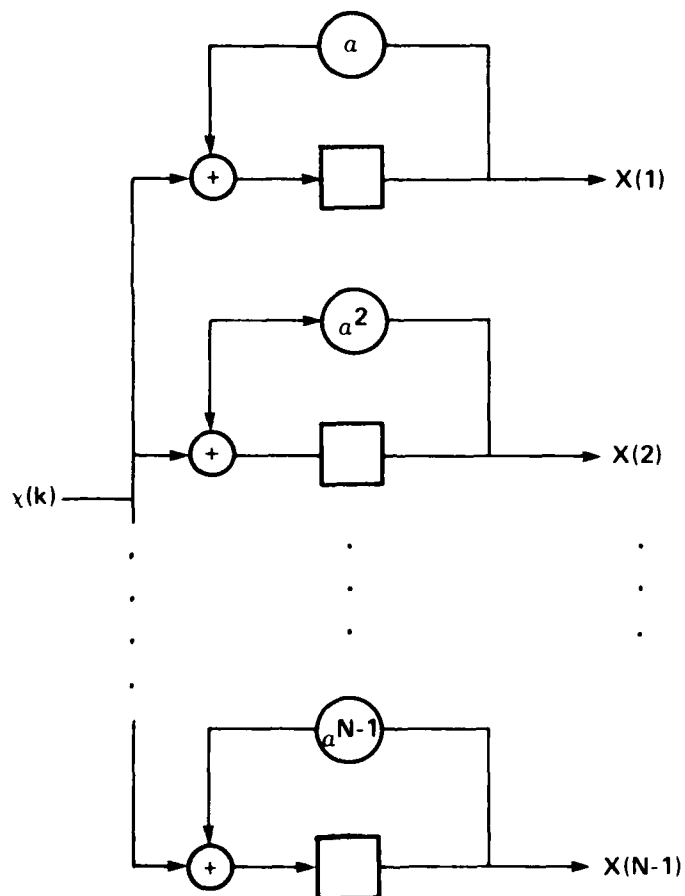
$$x^n - 1 = \prod_{j=0}^{n-1} (x - b^j) \quad (29)$$

the first degree factors $(x - b^j)$ being the modulus polynomials of equation (28). This factorization leads to the direct computation of transform values, requiring n^2 multiplications in $GF(p^m)$. A block diagram of a circuit to perform the calculation is shown in Figure 15.

Another factorization, one that reduces the number of multiplications in $GF(p^m)$, results from a successive decomposition of $x^n - 1$ into factors of the form $(x^{2^k} - b^{2^{\ell}})$. Observe that for $p > 2$ and $n = p^m - 1$ we can always establish that $b^{N/2} = -b^0$; therefore

$$(x^k - b^{\ell}) (x^k - b^{N/2 + \ell}) = (x^k - b^{\ell}) (x^k + b^{\ell}) = (x^{2k} - b^{2\ell}). \quad (30)$$

The polynomial $x^n - 1$ can be progressively factored in this manner, the factorization being represented conveniently as a binary tree, as shown in Figure 16 for $x^{80} - 1$ factored over $GF(3^4)$. It is easy to show that the evaluation at one of the roots b^j only requires processing along one of the distinct tree paths. This tends to reduce the number of



1A-59,240

Figure 15. FIRST DEGREE POLYNOMIAL DIVIDER STRUCTURE FOR AN N-POINT DISCRETE TRANSFORM:

$$X(\ell) = \sum_{k=0}^{N-1} x(k) \alpha^{k\ell}$$

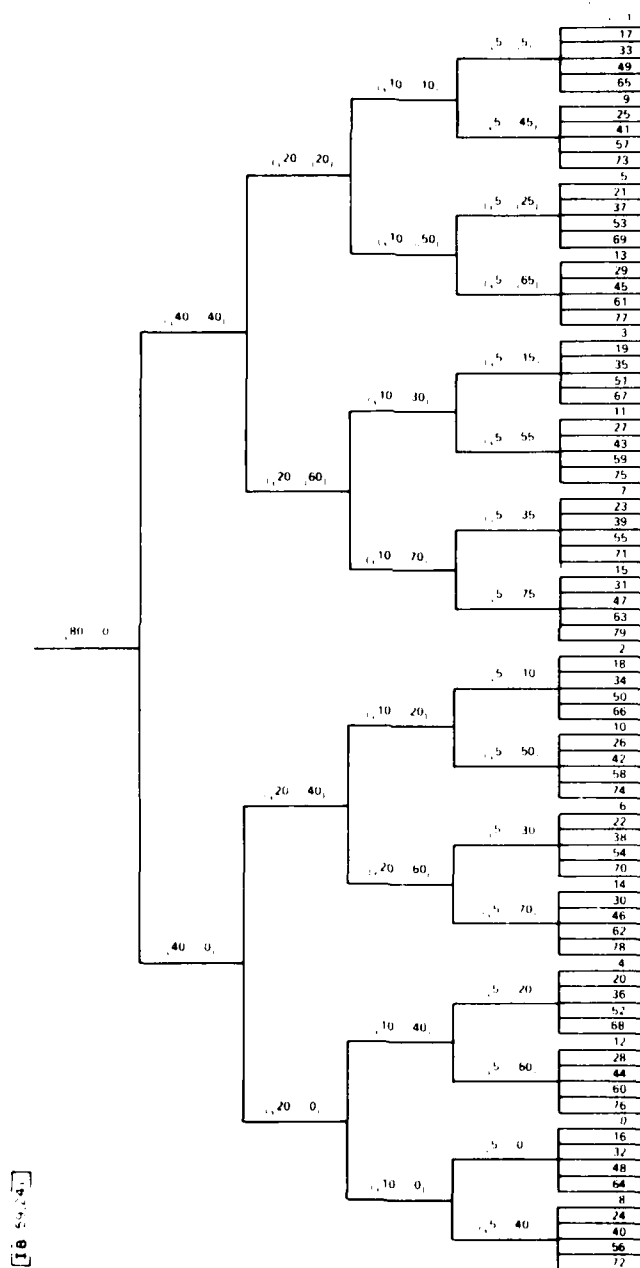


Figure 16. FACTORIZATION OF $x^{80}-1$ OVER $GF(3^4)$

multiplications in $GF(p^m)$ from n^2 to something on the order of $n \log_2 n$. Notice that we can write the division algorithms

$$a(x) = P_1(x) Q_1(x) + r_1(x) \quad (31a)$$

$$r_1(x) = P_2(x) Q_2(x) + r_2(x) \quad (31b)$$

and after substitution

$$a(x) = P_1(x) Q_1(x) + P_2(x) Q_2(x) + r_2(x). \quad (32)$$

If $P_2(x)$ divides $P_1(x)$, we can write equation (32) as

$$a(x) = P_2(x) \left[\frac{P_1(x)}{P_2(x)} Q_1(x) + Q_2(x) \right] + r_2(x) \quad (33)$$

which demonstrates that the remainder $r_2(x)$ can be calculated progressively by dividing $a(x)$ by $P_1(x)$ and then dividing the first remainder $r_1(x)$ by $P_2(x)$. The sequence can be continued indefinitely as we progress along a path in the tree. This type of decomposition is analogous to the decimation-in-frequency FFT algorithm.

The processing structure that accomplishes an 80-point transform over $GF(3^4)$ by use of this method is shown in Figure 17. There are 480 multiplications in $GF(3^4)$ required; of these approximately one-sixth are simple multiplications by $\pm b^0$.

A further reduction in the number of multiplications in $GF(p^m)$ required to calculate an n -point transform is possible by consideration of a different factorization of $x^n - 1$. If we factor this polynomial into the product of the minimal polynomials of the field elements, then we can devise a two-step algorithm in which the first step is division by the set of minimal polynomial factors and the second step is division of the remainder polynomials by the first degree polynomials $(x - b^j)$ that are factors of the minimal polynomials.

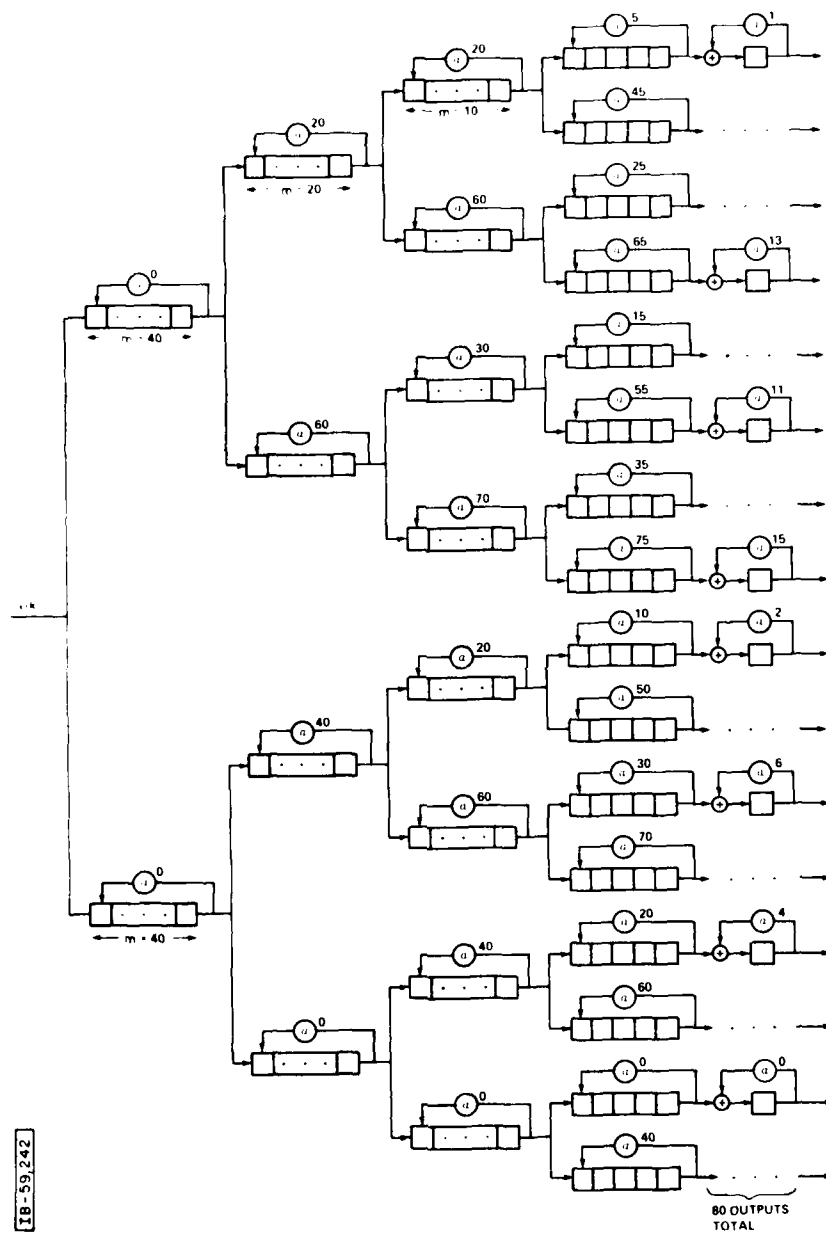


Figure 17. 80-POINT TRANSFORM OVER $GF(3^4)$
BY SPECTRAL DECIMATION (SCHEMATIC)

Explicitly, we can factor

$$x^n - 1 = \prod_{i=0}^M m_i(x) \quad (34)$$

where

$$m_i(x) = \prod_{j=1}^k (x - b^i p^j) \quad (35)$$

are the minimal polynomials. These are monic, irreducible over $GF(p)$, and have all coefficients in $GF(p)$. Division by these polynomials in the first step of the algorithm replaces multiplications in $GF(p^m)$ by multiplications in $GF(p)$ which are generally much simpler to perform. The second step of the algorithm requires multiplications in $GF(p^m)$ to evaluate the remainder polynomials at the points of the field, but the number of these multiplications is greatly reduced because there are a relatively small number of remainder polynomials, each of degree less than the degree of field extension. The number of multiplications in this final step could be further reduced, at the expense of more additions, by using the field recursion that expresses all the field elements as linear combinations of a subset of size m .

To illustrate this second algorithm, and compare it with the decimation-in-frequency type, we have worked out an example for an 80-point transform over $GF(3^4)$. In Table II, we list the elements of $GF(3^4)$ representing them as 4-tuples over the set $\{-1, 0, 1\}$ used to represent $GF(3)$. In Table III, we list the minimal polynomials of $GF(3^4)$ and their respective roots in $GF(3^4)$. In Figure 18 we show schematically a processing structure for calculating the transform. The circuits that divide the input data by the minimal polynomials are linear feedback shift registers over $GF(3)$. In this example, the number of multiplications in $GF(3^4)$ is 216, while 6800 multiplications in the base field are performed by 85 multipliers.

Table II
THE ELEMENTS OF $GF(3^4)$
GENERATED BY $\alpha^4 + \alpha^3 + \alpha^2 - \alpha - 1$

ELEMENT	REPRESENTATION	ELEMENT	REPRESENTATION
α^0	0 0 0 1	α^{40}	0 0 0-1
α^1	0 0 1 0	α^{41}	0 0-1 0
α^2	0 1 0 0	α^{42}	0-1 0 0
α^3	1 0 0 0	α^{43}	-1 0 0 0
α^4	-1-1 1 1	α^{44}	1 1-1-1
α^5	0-1 0-1	α^{45}	0 1 0 1
α^6	-1 0-1 0	α^{46}	1 0 1 0
α^7	1 0-1-1	α^{47}	-1 0 1 1
α^8	-1 1 0 1	α^{48}	1-1 0-1
α^9	-1 1 0-1	α^{49}	1-1 0 1
α^{10}	-1 1 1-1	α^{50}	1-1-1 1
α^{11}	-1-1 1-1	α^{51}	1 1-1 1
α^{12}	0-1 1-1	α^{52}	0 1-1 1
α^{13}	-1 1-1 0	α^{53}	1-1 1 0
α^{14}	-1 0-1-1	α^{54}	1 0 1 1
α^{15}	1 0 1-1	α^{55}	-1 0-1 1
α^{16}	-1 0 0 1	α^{56}	1 0 0-1
α^{17}	1 1 0-1	α^{57}	-1-1 0 1
α^{18}	0-1 0 1	α^{58}	0 1 0-1
α^{19}	-1 0 1 0	α^{59}	1 0-1 0
α^{20}	1-1-1-1	α^{60}	1 1 1 1
α^{21}	1 1 0 1	α^{61}	-1-1 0-1
α^{22}	0-1-1 1	α^{62}	0 1 1-1
α^{23}	-1-1 1 0	α^{63}	1 1-1 0
α^{24}	0-1-1-1	α^{64}	0 1 1 1
α^{25}	-1-1-1 0	α^{65}	1 1 1 0
α^{26}	0 0-1-1	α^{66}	0 0 1 1
α^{27}	0-1-1 0	α^{67}	0 1 1 0
α^{28}	-1-1 0 0	α^{68}	1 1 0 0
α^{29}	0 1-1-1	α^{69}	1-1 1 1
α^{30}	1-1-1 0	α^{70}	-1 1 1 0
α^{31}	1 1 1 1	α^{71}	-1-1-1-1
α^{32}	0 0-1 1	α^{72}	0 0 1-1
α^{33}	0-1 1 0	α^{73}	0 1-1 0
α^{34}	-1 1 0 0	α^{74}	1-1 0 0
α^{35}	-1 1-1-1	α^{75}	1-1 1 1
α^{36}	-1 0 1-1	α^{76}	1 0-1 1
α^{37}	1-1 1-1	α^{77}	-1 1-1 1
α^{38}	1 0 0 1	α^{78}	-1 0 0-1
α^{39}	-1-1-1 1	α^{79}	1 1 1-1

Table III
MINIMAL POLYNOMIAL FACTORS OF $x^{80} - 1$
(SPLITTING FIELD: $GF(3^4)$)

Polynomial	Roots: $\alpha^i, GF(3^4)$
$m_1(x) = x^4 + x^3 + x^2 - x - 1$	$\alpha, \alpha^3, \alpha^9, \alpha^{27}$
$m_2(x) = x^4 + x^3 + x^2 + 1$	$\alpha^2, \alpha^6, \alpha^{18}, \alpha^{54}$
$m_4(x) = x^4 - x^3 - x + 1$	$\alpha^4, \alpha^{12}, \alpha^{36}, \alpha^{28}$
$m_5(x) = x^4 + x^2 - 1$	$\alpha^5, \alpha^{15}, \alpha^{45}, \alpha^{55}$
$m_8(x) = x^4 + x^2 + 1$	$\alpha^8, \alpha^{24}, \alpha^{72}, \alpha^{56}$
$m_{10}(x) = x^2 + x - 1$	α^{10}, α^{30}
$m_{11}(x) = x^4 + x - 1$	$\alpha^{11}, \alpha^{33}, \alpha^{12}, \alpha^{57}$
$m_{13}(x) = x^4 - x^3 - x^2 + x - 1$	$\alpha^{13}, \alpha^{39}, \alpha^{37}, \alpha^{31}$
$m_{14}(x) = x^4 - x^3 + x^2 + 1$	$\alpha^{14}, \alpha^{42}, \alpha^{46}, \alpha^{58}$
$m_{16}(x) = x^4 + x^3 + x^2 + x + 1$	$\alpha^{16}, \alpha^{48}, \alpha^{64}, \alpha^{32}$
$m_{17}(x) = x^4 - x - 1$	$\alpha^{17}, \alpha^{51}, \alpha^{73}, \alpha^{59}$
$m_{20}(x) = x^2 + 1$	α^{20}, α^{60}
$m_{22}(x) = x^4 + x^2 - x + 1$	$\alpha^{22}, \alpha^{66}, \alpha^{38}, \alpha^{34}$
$m_{23}(x) = x^4 - x^3 - 1$	$\alpha^{23}, \alpha^{69}, \alpha^{47}, \alpha^{61}$
$m_{25}(x) = x^4 - x^2 - 1$	$\alpha^{25}, \alpha^{75}, \alpha^{65}, \alpha^{35}$
$m_{26}(x) = x^4 + x^2 + x + 1$	$\alpha^{26}, \alpha^{78}, \alpha^{74}, \alpha^{62}$
$m_{40}(x) = x + 1$	α^{40}
$m_{41}(x) = x^4 - x^3 + x^2 - 1$	$\alpha^{41}, \alpha^{43}, \alpha^{49}, \alpha^{67}$
$m_{44}(x) = x^4 - x^3 + x + 1$	$\alpha^{44}, \alpha^{52}, \alpha^{76}, \alpha^{68}$
$m_{53}(x) = x^4 + x^3 - x^2 - x - 1$	$\alpha^{53}, \alpha^{79}, \alpha^{77}, \alpha^{71}$
$m_{50}(x) = x^2 - x - 1$	α^{50}, α^{70}
$m_{80}(x) = m_0(x) = x - 1$	α^0
$m_7(x) = x^4 - x^3 - 1$	$\alpha^7, \alpha^{21}, \alpha^{63}, \alpha^{29}$

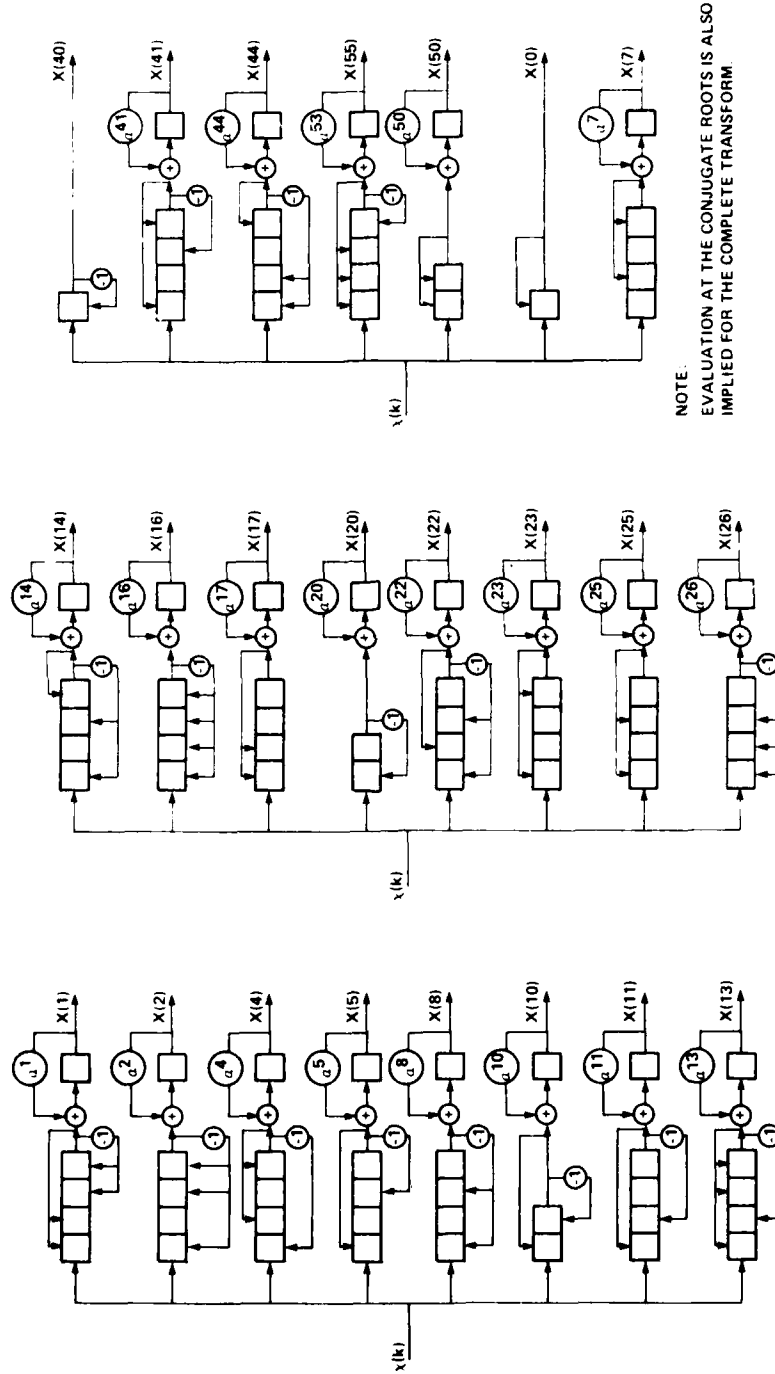


Figure 18, SCHEMATIC OF AN 80-POINT TRANSFORM OVER $GF(3^4)$
BY FAST POLYNOMIAL EVALUATION

Some of the well-known results of number theory can be used to enumerate the irreducible polynomials of degree d over $GF(p)$, which in turn allows us to enumerate the minimal polynomials of each degree and consequently the number of required multiplications in $GF(p^m)$. This allows us to assess the complexity of the algorithm for a number of different cases without explicitly determining the structure. The numbers of required multiplications over $GF(p^m)$ are enumerated in Table IV for a number of different cases, and the numbers are plotted in Figure 19 to compare the trend with the $N \log_2 N$ behavior of the FFT class of algorithms.

It is possible to pursue the idea of factoring $x^n - 1$ to devise a further simplification that reduces the number of multiplications over $GF(p)$ that need to be performed. In particular, we can first factor $x^n - 1$ into the product of the cyclotomic polynomials $Q^{(k)}(x)$ where the indices k are divisors of n . Thus, for example

$$x^{80} - 1 = Q^{(80)}(x) Q^{(40)}(x) Q^{(20)}(x) Q^{(16)}(x) Q^{(10)}(x) \dots \quad (36) \\ \dots Q^{(x)}(x) Q^{(5)}(x) Q^{(4)}(x) Q^{(2)}(x) Q^{(1)}(x)$$

The cyclotomic polynomials for this example are listed in Table V. In general, these polynomials have coefficients that are either 0, or ± 1 , up to index 105. The non-zero coefficients are typically sparse. Each of the polynomials $Q^{(d)}(x)$ can be factored over $GF(p^m)$ into the product of the minimal polynomials of the field elements of order d , so that the cyclotomic factorization, if it precedes the second fast algorithm described above, has the effect of reducing the number of multiplications in $GF(p)$ that are not multiplications by the set $\{0, 1, -1\}$ (which are also the elements of $GF(3)$ by coincidence). The factorization of $x^n - 1$ into cyclotomic polynomials depends only on n and is otherwise independent of the field over which the transform is being calculated. The processor structure for this first step can therefore be field-independent. As an example of the

TABLE IV
MULTIPLICATIVE COMPLEXITY OF DISCRETE
TRANSFORM ALGORITHM BY FAST POLYNOMIAL EVALUATION

FIELD	NUMBER OF TRANSFORM POINTS (N)	NUMBER OF EXTENSION FIELD PRODUCTS	$N \log_2 N$
$GF(2^5)$	31	120	154
$GF(2^7)$	127	756	1,214
$GF(2^8)$	255	1,719	2,038
$GF(2^{11})$	2047	20,461	22,517
$GF(3^3)$	28	50	122
$GF(3^4)$	80	224	505
$GF(3^5)$	242	962	1916
$GF(3^7)$	2186	13,106	24,253
$GF(5^2)$	24	24	110
$GF(5^3)$	124	247	862
$GF(5^5)$	3124	12,484	36,267

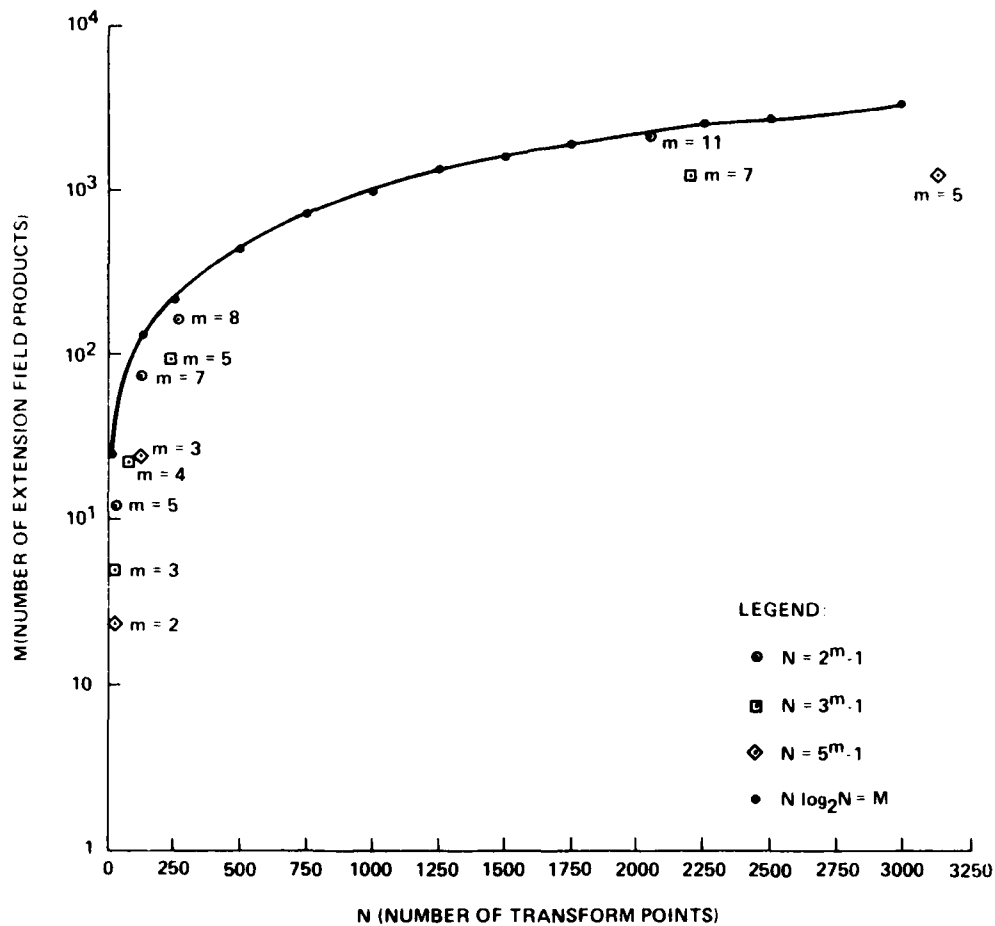


TABLE V
CYCLOTOMIC FACTORS OF $x^{80} - 1$

CYCLOTOMIC POLYNOMIAL	IRREDUCIBLE FACTORS*
$Q^{(1)}(x) = x-1$	$m_{80}(x) = m_0(x)$
$Q^{(2)}(x) = x+1$	$m_{40}(x)$
$Q^{(4)}(x) = x^2+1$	$m_{20}(x)$
$Q^{(5)}(x) = x^4+x^3+x^2+x+1$	$m_{15}(x)$
$Q^{(8)}(x) = x^4+1$	$m_{10}(x), m_{50}(x)$
$Q^{(10)}(x) = x^4-x^3+x^2-x+1$	$m_8(x)$
$Q^{(16)}(x) = x^8+1$	$m_5(x), m_{25}(x)$
$Q^{(20)}(x) = x^8-x^6+x^4-x^2+1$	$m_4(x), m_{44}(x)$
$Q^{(40)}(x) = x^{16}-x^{12}+x^8-x^4+1$	$m_2(x), m_{14}(x), m_{22}(x), m_{26}(x)$
$Q^{(80)}(x) = x^{32}-x^{24}+x^{16}-x^8+1$	$m_1(x), m_7(x), m_{11}(x), m_{17}(x),$ $m_{23}(x), m_{13}(x), m_{41}(x), m_{53}(x)$

* See Table III.

use of these polynomials, we have shown in Figure 20 a set of dividers that can precede the structure of Figure 18. In this case, there is only a slight decrease in the number of multiplications in $GF(3)$. As a second example, we show in Figure 21 the complete structure for a 24-point transform in $GF(5^2)$. In this case, the number of multiplications in $GF(5)$ that are not multiplications by $\{0,1,-1\}$ has been reduced from 218 to 64 by incorporating the cyclotomic factorization step.

3.4 Cyclic Convolution

The successful application of CCD multi-level logic to digital signal processing will depend, among other things, on the ability to devise means of utilizing CCD structures that are either relatively easy to fabricate or are minor modifications of existing devices. Structures such as tapped delay lines and programmable transversal filters fall into this category, but they have certainly not been developed in anticipation of multiple-valued digital operation such as we envision. Below we discuss an idea for utilizing the generic binary-programmable transversal filter structure for performing finite-field cyclic convolution. In later sections, we will expand the idea for other applications.

Convolution is a frequently encountered signal processing operation. The cyclic convolution of two n -point sequences with elements belonging to $GF(p)$ can be regarded as the product, modulo x^n-1 , of two polynomials of degree $n-1$ having coefficients in $GF(p)$. It seems likely that the type of binary-programmable transversal filter useful for PN-sequence matched filtering, developed for correlating an analog signal against a stored binary reference, can also be used for finite-field convolution. Just as it is possible to perform an analog-analog correlation by A/D conversion of the reference followed by parallel correlation in several analog-binary devices, so is it possible to partition the finite-field operations among a set of elementary correlators.

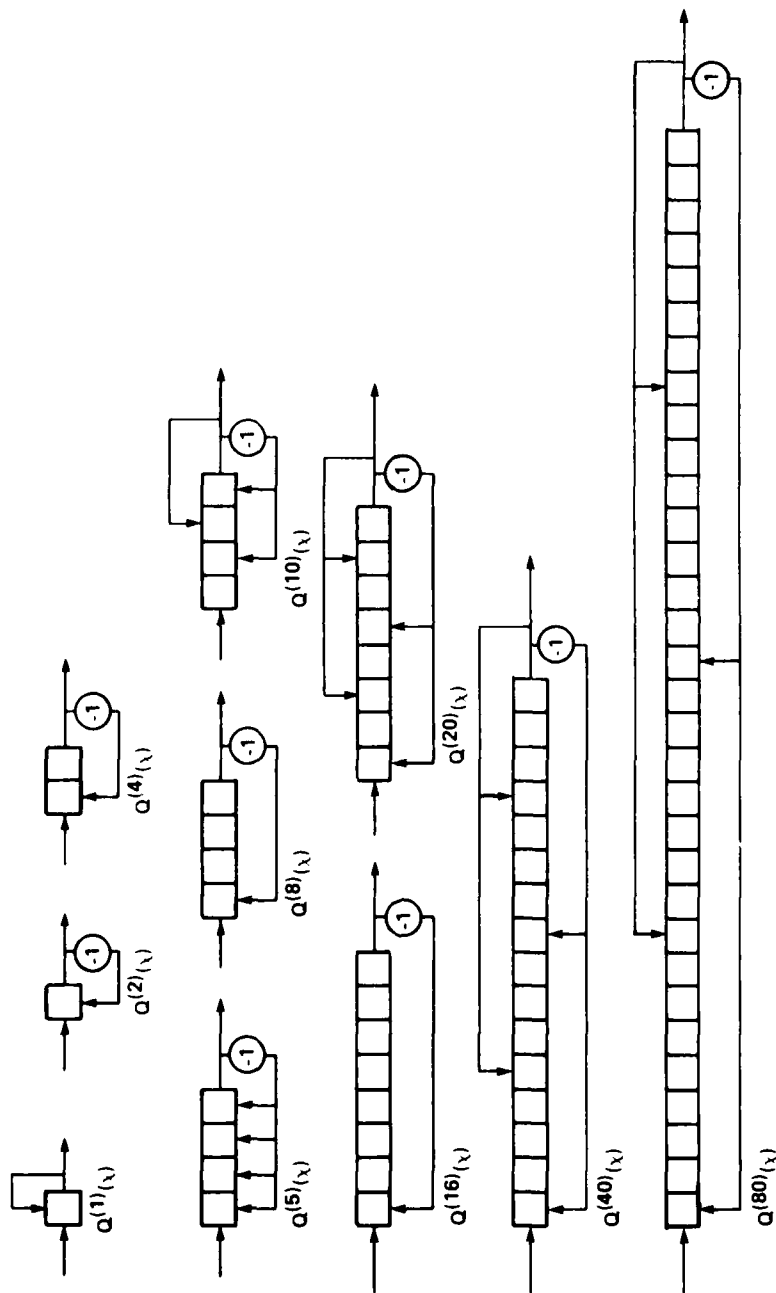


Figure 20, POLYNOMIAL DIVIDERS FOR THE CYCLOTOMIC FACTORS OF $x^{80}-1$

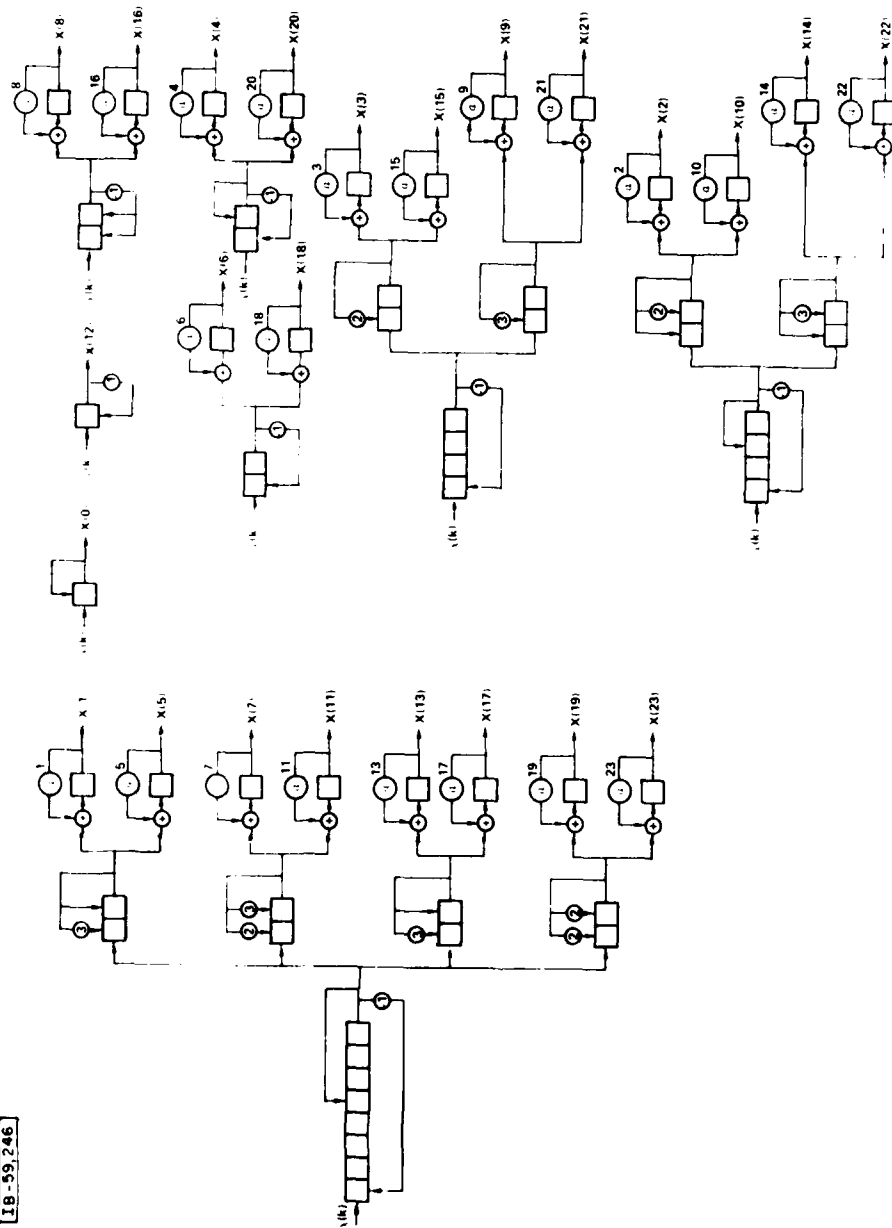


Figure 21. 24-POINT TRANSFORM OVER $GF(5^2)$

If we represent the elements of $GF(p)$ by the set $\{-(p-1)/2, \dots, -1, 0, +1, \dots, +(p-1)/2\}$ then we can treat the multiplication of an element b by another element a by the elementary process of summing b to itself a times. For cyclic convolution, we may use such a representation to form the various products over $GF(p)$ in a set of tapped delay lines that apply the tap weights, $0, \pm 1$ and accumulate the partial product in each component device before combining their outputs. Of course the accumulated sum must be reduced modulo p for polynomial multiplication but this can be done separately at the output of each correlator, as well as at the final output, in order to limit the dynamic range.

A structure that correlates a sequence over $GF(5)$ with the m -sequence generated by the primitive polynomial $\alpha^2 + \alpha + 2$ is shown schematically in Figure 22. The m -sequence is also shown for reference. In this diagram it is assumed that the zero-value of the signal is represented by a charge value Q_0 that is at the mid-range of a full well. Signal charges weighted by $+1$ are routed to the positive summing bus while those weighted by -1 are routed to the negative summing bus. For a zero tap weight the signal charge is not routed to either bus. We assume, of course, that the charge sensing is nondestructive.

An application suggested by the apparatus of Figure 22 is a matched filter detector for PN sequences defined over $GF(p)$. In such an application, the modulo p reduction is not needed as the cross-correlation is formed in the ordinary number field. The autocorrelation function of the m -sequence used is shown in Figure 23. In this figure, we have also shown the cross-correlation of the m -sequence with the reference used in the central correlator, which is simply a hard-limited version of the m -sequence. We see that the full correlation produces an improvement of 8 db relative to the hard-limited version (the output of the central correlator), as well as suppressing the positive sidelobes.

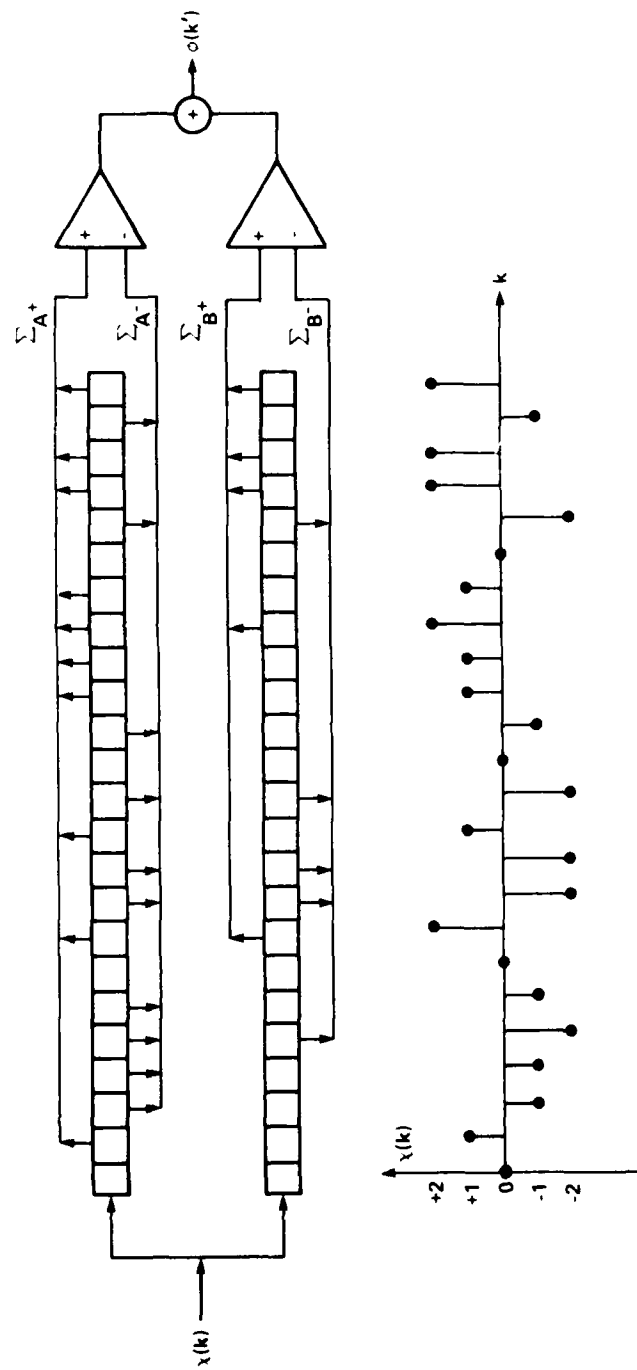


Figure 22. PARTITIONED TRANSVERSAL CORRELATOR FOR
A 24-POINT M-SEQUENCE OVER GF(5) GENERATED
BY $x^2 + x + 2$

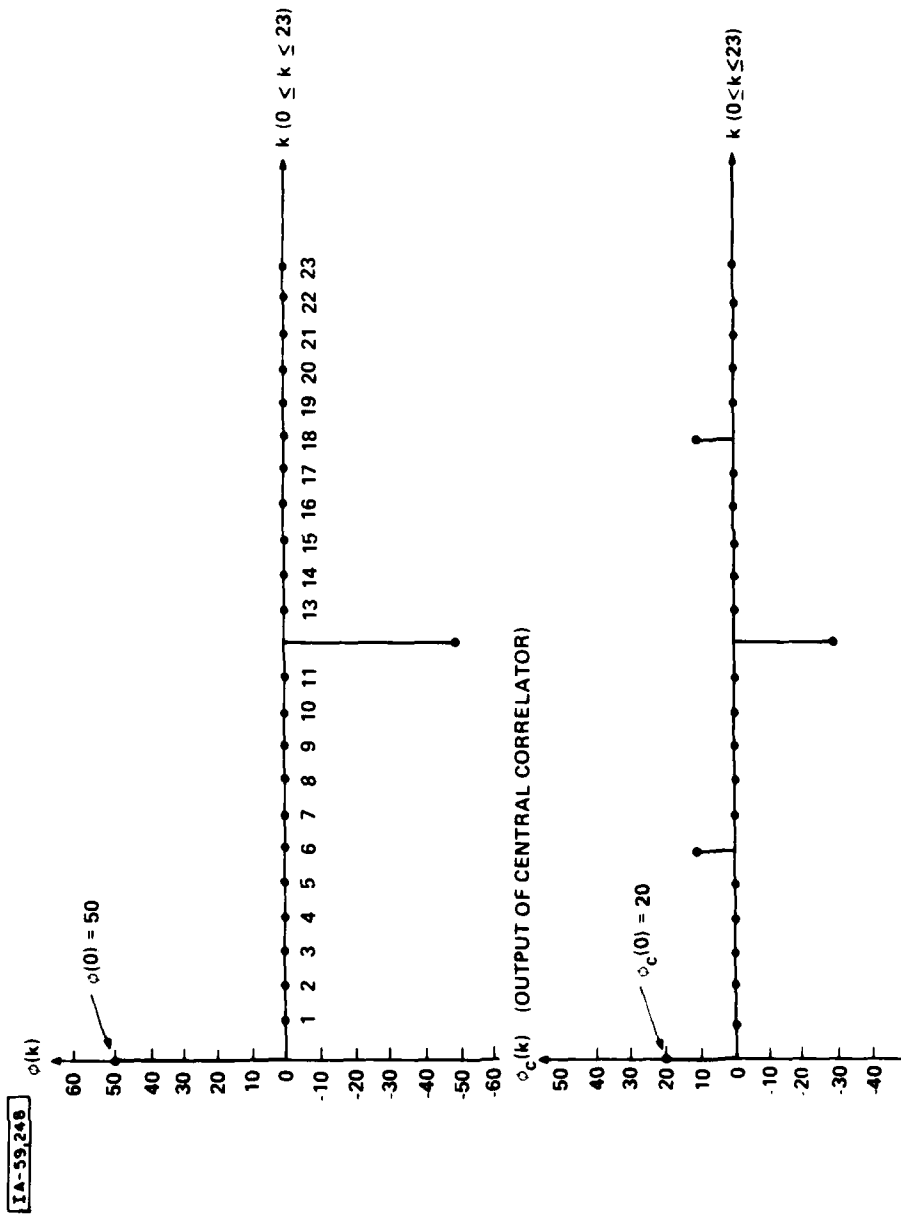


Figure 23. CYCLIC AUTOCORRELATION OF M-SEQUENCE OVER
GF(5) GENERATED BY $x^2 + x + 2$

An additional refinement to the apparatus could be made if it is not desired to represent the zero-value of the signal by a positive charge. In that case, positive and negative signal samples could be detected and separately correlated in sets of correlator-banks, each bank operating only on positive signal samples and the zero-value being represented by the bias charge (fat zero) setting the minimum charge level for the wells. This approach would require twice as many correlators, but the dynamic range requirements would be relaxed somewhat.

3.5 Polynomial Division with Transversal Structures

As we have shown in Section 3.3, division by polynomials over $GF(p)$ is an important step in processing to calculate the discrete transform of an input sequence defined over $GF(p^m)$. The polynomial dividers described in that section implemented the division algorithm

$$A(x) = P(x) Q(x) + R(x) \quad (37)$$

to divide the polynomial $A(x)$ by the polynomial $P(x)$. It was assumed that $A(x)$ was a polynomial over $GF(p^m)$ while the divider polynomial $P(x)$ was defined over $GF(p)$, the division being carried out simultaneously by a set of m identical linear feedback shift registers. It will suffice to consider just one of these shift registers, treating its input as a sequence over $GF(p)$. It is appropriate then to write the division algorithm as

$$a(x) = P(x) q(x) + r(x) \quad (38)$$

where all of the polynomials are defined over $GF(p)$. The m -ary case is carried out by m such divisions in parallel.

An inconvenience for CCD implementation of the division register used to implement equation (38) is that charge summation is required in certain stages determined by the divider polynomial. This has

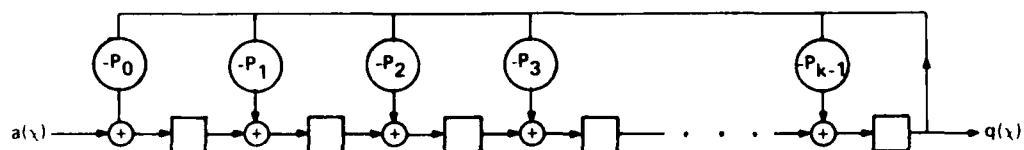
several drawbacks. First of all the summation must be defined modulo p if an excessive buildup of charge along the register is to be avoided. This complicates the structure of the shift register. Although modulo p adders can be designed, as shown conceptually in Section 3.2, it seems preferable to separate the adder from the delay line. In that case, a transversal structure seems more appropriate.

The polynomial divider that implements equation (38) may be regarded as a digital filter whose input is the polynomial $a(x)$ and whose output is the quotient $q(x)$. The feedback taps are determined by the divisor $P(x)$ and the remainder $r(x)$ is left in the register after the input $a(x)$ has been processed. The filter can be transposed into a transversal form by using signal flow-graph techniques (reverse all paths, exchange adders and path nodes, exchange input and output nodes). The transversal filter implements the same input-output function as the original divider; in other words it has the same unit pulse response. An example of a divider-network filter and its transposed version is shown in Figure 24. Although the two filters have the same unit pulse response, the circuit state as represented by the register contents differs on each cycle. For our application of polynomial division, it is the remainder polynomial $r(x)$ representing the final state that is of principal interest so the transversal filter structure cannot be used directly.

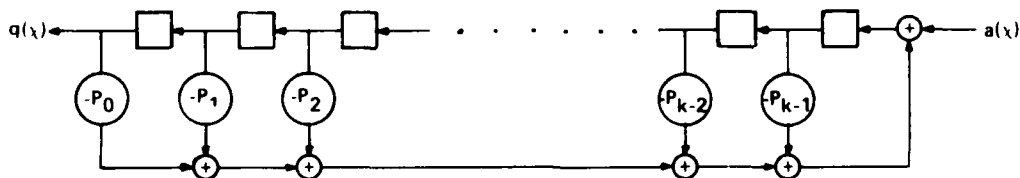
We may, however, use the transversal filter approach to polynomial division to determine the remainder in two steps; the first step is to determine the quotient $q(x)$ and the second step is used to calculate

$$r(x) = a(x) - P(x) q(x) \quad (39)$$

which is the needed result. The transposed divider circuit is used to find the quotient which is then multiplied by the divider



(a) DIVIDER FOR $P(x) = P_{k-1}x^{k-1} + P_{k-2}x^{k-2} + \dots + P_0$



(b) TRANSPOSED DIVIDER

14-59249

Figure 24. POLYNOMIAL DIVIDER AND ITS TRANSPOSED (TRANSVERSAL) FORM

polynomial in a second transversal filter and the product is subtracted from the suitably delayed input sequence. A transversal structure that implements division of a 24-point sequence by the cyclotomic polynomial $Q^{(12)}(x)$ is shown in Figure 25 as an example of the method. In comparison with the canonic LFSR divider, additional circuitry is required, as well as additional processing time, but the tapped delay line transversal structure seems more convenient to implement with available CCD techniques making the tradeoff a reasonable one.

3.6 Galois Field Representation With m-Sequences

A finite field processing operation that arises frequently is multiplication of pairs of elements of $GF(p^m)$. For example, this operation was a major concern in the development of a fast algorithm for calculating an N-point discrete transform in $GF(p^m)$ as discussed in Section 3.3. One method of performing the multiplication is to use a linear sequential circuit designed over $GF(p)$ to multiply a data sample α^l by a constant α^k , the feedback and feedforward connections being determined by the scale factor and the data providing the initial loading of the shift register. The product is formed by shifting the register once. This technique was described in Section 3.1. Although the operations required are additions and multiplications in the prime field $GF(p)$ rather than in the extension field $GF(p^m)$, it was evident that a number of adders and multipliers are required and that the sequential circuit structure is quite different for each scale factor.

If both the field characteristic and degree of extension are small, then the possibility presents itself of representing the field elements by distinct cyclic shifts of the m-sequence generated by the primitive generator of the field. This can have dramatic effect in reducing hardware complexity for implementing the arithmetic operations at the expense of increased sequential processing. The

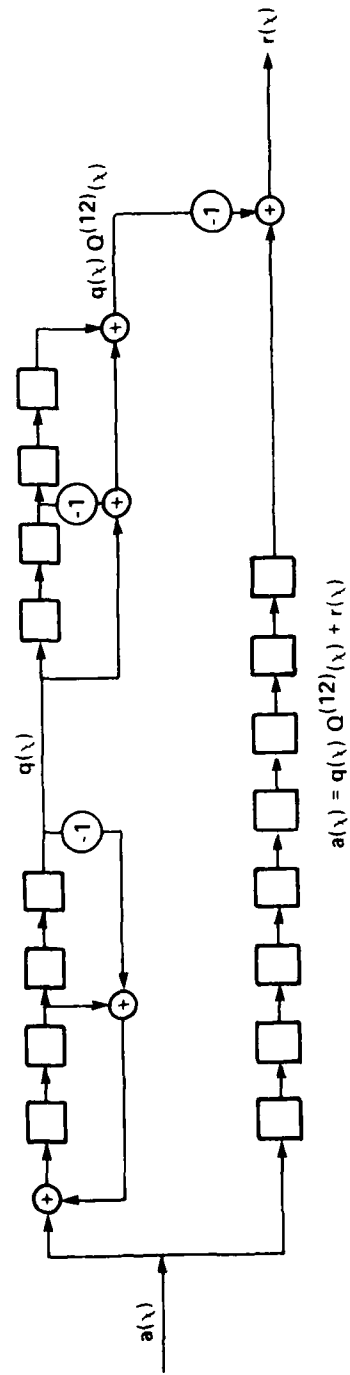


Figure 25. TRANSVERSAL POLYNOMIAL DIVIDER FOR
DIVISION BY $q(x) = x^4 - x^2 + 1$

time-expansion factor for sequential multiplication is $(p^m-1)/m$ which practically restricts the representation to small values of both p and m .

The transformation $(h: \alpha \rightarrow \beta)$ that maps the elements of $GF(p^m)$ into cyclically shifted m -sequences is a bijective mapping that maps the identity element into itself for both the additive and multiplicative groups of the field, thus causing the group transformations to be group isomorphisms. The practical consequences of this algebraic statement are that multiplication of two elements β^i and β^j is accomplished by cyclically shifting the element $(\beta^i), j$ - times (or the element $(\beta^j), i$ - times), and that addition of β^i and β^j is accomplished by the component-wise sum, modulo p , of their (p^m-1) -tuple representations. For addition of the elements β^i , as in the case of the elements α^i , no carry operations are required.

Multiplication of a data sample α^i by a constant factor α^k can be performed by circularly shifting the data in a recursive loop $N = p^m - 1$ times while reading out the product at a tap determined by the multiplier α^k as shown in Figure 26. In this figure, we also show the addition of another data sample α^l to the product to implement the first degree function $ax_1 + x_2 = \alpha^k \alpha^i + \alpha^l$. The adder shown must be a modulo p adder, but only one of these is required since the function is formed sequentially.

A pair of circuits of the type shown in Figure 26 could be interconnected to operate alternately, to performing the function of polynomial evaluation. This operation is important in the computation of the discrete transform even when a fast algorithm is employed, as discussed in Section 3.3.

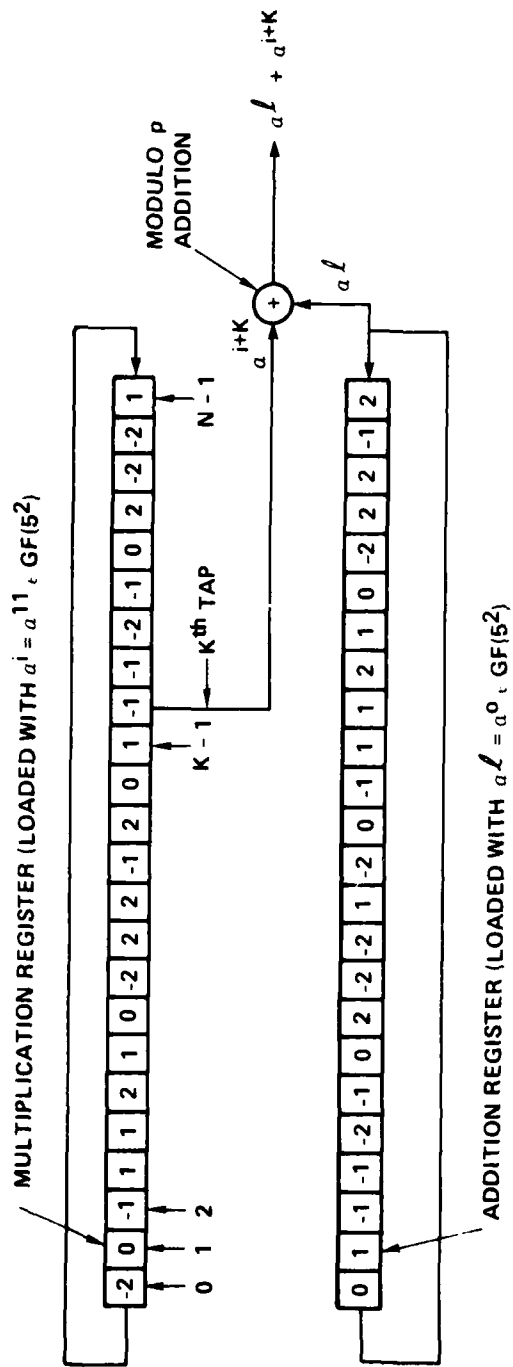


Figure 26. FINITE FIELD REPRESENTATION BY M-SEQUENCE:
SCALAR MULTIPLICATION AND ADDITION

The output of the polynomial circuit of Figure 2c is in the m-sequence representation and needs to be transformed back to the compressed representation eventually, although further computations can be performed in the m-sequence represented field. A number of methods suggest themselves for the inverse transformation, but the correlation detection approach seems most appropriate, and raises the distinct possibility of incorporating a degree of fault tolerance into the operation. A method of correlation detection was discussed above in Section 3.4.

The m-sequence representation was introduced with the motive of reducing hardware complexity while utilizing generic CCD functions based on shift register operations and transversal structures. It is apparent that some of the operations are relevant both to error-coding and to spread-spectrum matched filtering. That raises the distinct possibility that in systems which combine these signal processing functions (such as JTIDS), it may be possible to utilize the m-sequence representation to combine some of the processing functions or hardware elements, or both, used for PN sequence demodulation and Reed-Solomon decoding; and perhaps it could be used to inject a degree of fault tolerance into the hardware. The subject merits further exploration as an area of application for some of the techniques discussed above.

REFERENCES

1. Roome, T. F. "Generalized Cyclic Codes Finite Field Arithmetic," ESD-TR-79-124, Electronic Systems Division, AFSC, Hanscom AFB, MA, May 1979, ADA070673.
2. Ellison, J. T. and Cohn, M., "Fault Tolerance in Galois Linear Arrays." SPERRY UNIVAC Report PX-7931, Defense Systems Div., St. Paul, Minn.: October 1972.
3. Zimmerman, T. A., "The Digital Approach to Charge Coupled Device Signal Processing", IEEE Advanced Solid-State Components for Signal Processing, IEEE International Symposium on Circuits and Systems, April 1975 (IEEE Catalog No. 75 CH0979-5 CAS).
4. Pollard, J. M., "The Fast Fourier Transform in a Finite Field", Mathematics of Computation, Vol. 25, Number 114, April, 1971, pp. 365 - 374.
5. Murakami, H. and Reed, I. S., "Recursive Realization of Finite Impulse Filters Using Finite Field Arithmetic", IEEE Transactions on Information Theory, March 1977, pp. 232 - 242.
6. Carhoun, D.O., Roome, T.F., Palo, E.A., "Error Correction Coding with Charge Transfer Devices", MTP-176, The MITRE Corporation, Bedford, Mass., November 1976.
7. Carhoun, D.O., Roome, T.F., Palo, E.A., "Finite Field Arithmetic with Charge Transfer Devices", MTP-175, The MITRE Corporation, Bedford, Mass., November 1976.

END

DATE
FILMED

5 81

DTIC